

АВТОРСКИЙ КУРС



ЛАДА ЕСАКОВА

ИНФОРМАТИКА

ГОТОВИМСЯ К ЕГЭ С ЛУЧШИМИ
РЕПЕТИТОРАМИ РОССИИ

Авторский курс
подготовки к

ЕГЭ

ЕГЭ

Подготовка к ЕГЭ и олимпиадам
Образовательная компания ЕГЭ-Студия

ФЕНИКС

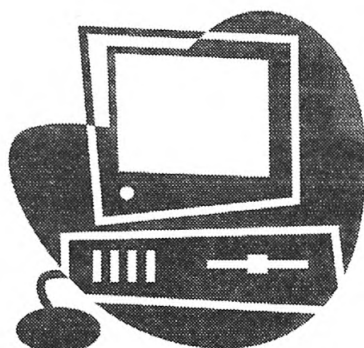
5+

Серия
«Авторский курс»

Л. Б. Есакова

ИНФОРМАТИКА

**Авторский курс
подготовки к ЕГЭ**



Издание 2-е

Ростов-на-Дону
«Феникс»
2018

УДК 373.167.1:004

ББК 32.81я72

КТК 448

Е81

Есакова Л. Б.

Е81 Информатика : авторский курс подготовки к ЕГЭ / Л. Б. Есакова. — Изд. 2-е. — Ростов н/Д : Феникс, 2018. — 283, [1] с. : ил. — (Авторский курс).

ISBN 978-5-222-30341-2

Пособие содержит подробный разбор всех типов задач, рекомендации по решению, а также краткие теоретические справки. Пособие предназначено для выпускников, планирующих сдавать ЕГЭ по информатике, а так же для их учителей.

Книга написана простым и понятным языком, без использования сложных научных терминов, и поможет эффективно подготовиться к экзамену ученикам разного уровня подготовки. Способы решения задач, предложенные в книге, зарекомендовали себя, как наиболее легко осваиваемые и позволяющие избежать случайных ошибок.

Пособие составлено на основании Демонстрационного варианта контрольных измерительных материалов единого государственного экзамена 2016 года по информатике и ИКТ. Также рассмотрены все основные типы задач, которые встречались в тренировочных, репетиционных и диагностических работах, ЕГЭ по информатике основной и досрочной волны 2013–2015 гг.

УДК 373.167.1:004

ББК 32.81я72

ISBN 978-5-222-30341-2

© Есакова Л. Б., 2017

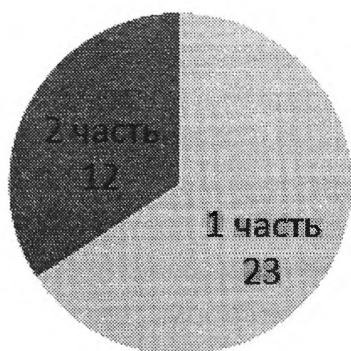
© Оформление: ООО «Феникс», 2018

Введение

В 2016 году структура экзаменационной работы ЕГЭ по информатике осталась практически без изменений (немного изменен порядок заданий 1-5, но количество заданий и выставяемые за них баллы не изменились). Давайте рассмотрим структуру экзаменационной работы. Всего в работе 27 заданий. Из них 23 с кратким ответом (бывшая часть В) и 4 с развернутым ответом (бывшая часть С). В экзаменационной работе, с одной стороны, присутствуют задания, требующие хорошего знания математики, логики, умения анализировать и абстрактно мыслить, с другой стороны, задания, основанные на аккуратном, монотонном исполнении алгоритма, переборе вариантов (ученику предоставляется возможность поработать, как компьютер).

Задания с кратким ответом оцениваются в 1 балл, задания с развернутым ответом: задача 24 — 3 балла, задача 25 — 2 балла, задача 26 — 3 балла, задача 27 — 4 балла. Таким образом, за первую часть можно получить максимально 23 балла, а за вторую 12.

Распределение баллов на ЕГЭ по информатике



Видите, насколько велик вес 1-й части?

В спецификации Единого государственного экзамена по ИНФОРМАТИКЕ и ИКТ указано рекомендованное время на выполнение каждого задания. На 1-ю часть рекомендовано потратить 1,5 часа (включая проверку и переписывание на бланк), а на 2-ю часть остается 2,5 часа. Это очень правильная рекомендация. Если планируешь сделать всю 2-ю часть, больше часа на решение 1-й части тратить нельзя (еще полчаса уйдут на проверку и переписывание)!

Но что такое час на 23 задания? Правильно, это меньше 3 минут на каждое! Задания 1-й части довольно простые для большей части учеников, но они часто требуют аккуратного перебора и анализа большого объема данных. Даже в случае отличного понимания предмета требования довольно жесткие! Поэтому очень важна практика, наработанный опыт решения всех типов задач.

Экзаменационная работа ЕГЭ охватывает основные темы курса информатики и ИКТ: «Информация и ее кодирование», «Моделирование и компьютерный эксперимент», «Системы счисления», «Логика и алгоритмы», «Элементы теории алгоритмов», «Языки программирования», «Архитектура компьютеров и компьютерных сетей», «Обработка числовой информации», «Технологии поиска и хранения информации».

В книге задания разбиты по темам на основании Кодификатора и Спецификации контрольных измерительных материалов для проведения в 2016 г. Единого государственного экзамена по информатике и ИКТ, подготовленных Федеральным государственным бюджетным научным учреждением «Федеральный институт педагогических измерений».

В книге рассмотрены все основные типы задач, которые встречались в тренировочных, репетиционных и диагностических работах, ЕГЭ по информатике основной и досрочной волны 2013–2015 гг., а также разобраны все задачи демонстрационной версии ЕГЭ по информатике 2016 г.

Данная книга не является учебником и содержит лишь краткие теоретические справки, необходимые для решения конкретной задачи.

Каждый из способов решения задач, предложенный в книге, не является единственно правильным, но рекомендуется автором как наиболее понятный, наглядный и дающий возможность избежать ошибок.

Задача № 1.

Системы счисления.

Кодирование чисел в разных системах

Система счисления — это способ представления чисел с помощью символов (цифр).

Системы счисления бывают *позиционные* и *непозиционные*.

Позиционная система счисления — это система, в которой значение числа зависит от положения цифры в записи этого числа. (Например, десятичные числа 73 и 37 различны, т.к. различны положения цифр 3 и 7.)

Непозиционная система счисления — это система, в которой значение числа не зависит от положения цифры в записи этого числа.

Мы не будем рассматривать непозиционные системы счисления в данном курсе, т.к. в современной жизни они практически не применяются.

Алфавит системы счисления — это набор символов (цифр), используемых для обозначения числа.

Например, в привычной нам десятичной системе алфавит состоит из цифр от 0 до 9, их количество 10, именно поэтому система называется десятичной.

Аналогично, в двоичной системе счисления алфавит состоит из двух цифр: 0 и 1. В 8-ричной — из восьми цифр и т.д.



Если количество символов в алфавите больше 10, добавляют буквы латинского алфавита. Например, в 16-ричной системе счисления алфавит следующий:

$$0, 1, 2, 3, 4, 5, 6, 7, 8, 9, \\ A(10), B(11), C(12), D(13), E(14), F(15).$$

Основанием системы счисления может быть любое натуральное число. Оно обозначается нижним индексом справа от числа. Например: 12_8 , 43_5 , 1101_2 .

Правило последовательного счета в различных системах счисления одинаково.

Для получения натурального числа, следующего за данным, необходимо самую правую цифру числа заменить на следующую за ней в алфавите (увеличить на 1), если же эта цифра последняя в алфавите, ее следует заменить на 0, а стоящую перед ней цифру увеличить на 1 и т.д. Если все цифры числа равны последней в алфавите цифре, то они заменяются на 0, а перед ними приписывается 1 (увеличивается разрядность числа).

Применяя это правило, запишем числа от 0 до 20 в нескольких системах счисления (получим таблицу соответствия записей чисел от 1 до 20 в различных системах):

N_{10}	N_2	N_3	N_5	N_8	N_{16}
0	0	0	0	0	0
1	1	1	1	1	1
2	10	2	2	2	2
3	11	10	3	3	3
4	100	11	4	4	4
5	101	12	10	5	5
6	110	20	11	6	6
7	111	21	12	7	7

N_{10}	N_2	N_3	N_5	N_8	N_{16}
8	1000	22	13	10	8
9	1001	100	14	11	9
10	1010	101	20	12	A
11	1011	102	21	13	B
12	1100	110	22	14	C
13	1101	111	23	15	D
14	1110	112	24	16	E
15	1111	120	30	17	F
16	10000	121	31	20	10
17	10001	122	32	21	11
18	10010	200	33	22	12
19	10011	201	34	23	13
20	10100	202	40	24	14

Перевод чисел в десятичную систему счисления из любой другой

Для перевода числа в десятичную систему из любой другой следует пронумеровать разряды числа начиная с нуля справа налево от младшего целого разряда и представить число в виде суммы произведений его цифр на основание системы в степени разряда числа.

Если в числе есть дробная часть, ее разряды нумеруются слева направо от -1 в убывающем порядке.

Пример 1.1

Переведите число 1042 из пятеричной системы в десятичную.

Решение

Пронумеруем разряды числа справа налево:

$$\begin{array}{r} 3 \ 2 \ 1 \ 0 \\ 1 \ 0 \ 4 \ 2_5 \end{array}$$

Посчитаем сумму произведений цифр числа на 5 (основание системы) в степени разряда числа:

$$1042_5 = 2 \cdot 5^0 + 4 \cdot 5^1 + 0 \cdot 5^2 + 1 \cdot 5^3 = 2 + 20 + 0 + 125 = 147_{10}.$$

Ответ: 147_{10} .

Пример 1.2

Переведите число $105,4$ из восьмеричной системы в десятичную.

Решение

Пронумеруем целые разряды числа справа налево от 0, дробные — слева направо от -1 :

$$\begin{array}{r} 2 \ 1 \ 0 \ -1 \\ 1 \ 0 \ 5, 4_8 \end{array}$$

Посчитаем сумму произведений цифр числа на 8 (основание системы) в степени разряда числа:

$$4 \cdot 8^{-1} + 5 \cdot 8^0 + 0 \cdot 8^1 + 1 \cdot 8^2 = 0,5 + 5 + 0 + 16 = 21,5_{10}.$$

Ответ: $21,5_{10}$.

Пример 1.3

Переведите двоичное число 101001_2 в десятичную систему.

$$1) 29_{10} \quad 2) 41_{10} \quad 3) 51_{10} \quad 4) 81_{10}$$

Решение

$$\begin{aligned} 101001_2 &= 1 \cdot 2^0 + 0 \cdot 2^1 + 0 \cdot 2^2 + 1 \cdot 2^3 + 0 \cdot 2^4 + 1 \cdot 2^5 = \\ &= 1 + 8 + 32 = 41_{10}. \end{aligned}$$

Ответ: 2.

Пример 1.4

Какое из неравенств выполняется для чисел $A = 164_8$, $B = A3_{16}$ и $C = 2200_4$?

1) $A < B < C$

2) $A < C < B$

3) $B < A < C$

4) $C < B < A$

Решение

$$A = 164_8 = 4 \cdot 8^0 + 6 \cdot 8^1 + 1 \cdot 8^2 = 4 + 48 + 64 = 116_{10}.$$

$$B = A3_{16} = 3 \cdot 16^0 + A \cdot 16^1 = 3 + 10 \cdot 16 = 163_{10}.$$

$$C = 2200_4 = 0 \cdot 4^0 + 0 \cdot 4^1 + 2 \cdot 4^2 + 2 \cdot 4^3 = 32 + 128 = 160_{10}.$$

Значит, $A < C < B$.

Ответ: 2.

Перевод целых чисел из десятичной системы счисления в любую другую

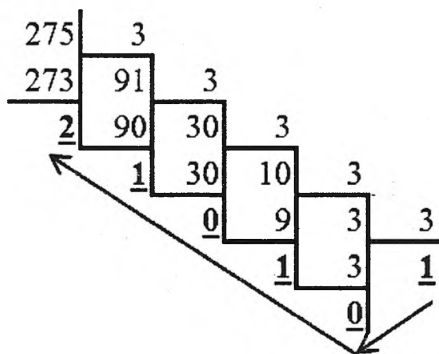
Для перевода числа из десятичной системы счисления в любую другую следует выполнять деление числа, а затем его целых частных на основание новой системы счисления до тех пор, пока частное не станет меньше делителя, затем остатки от деления записать начиная с последнего.

Пример 1.5

Переведите число 275 из десятичной системы в троичную.

Решение

Разделим число и его целые частные на 3:



Запишем остатки от деления начиная с последнего:

101012_3 .

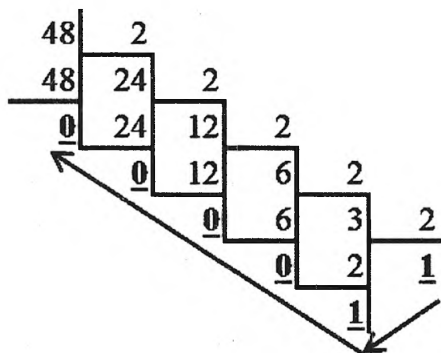
Ответ: 101012_3 .

Пример 1.6

Сколько значащих нулей в двоичной записи десятичного числа 48?

- 1) 1 2) 2 3) 6 4) 4

Решение



$$48_{10} = 110000_2.$$

Незначащими являются нули, стоящие перед числом (перед первой ненулевой цифрой), их можно отбросить. Например, в числе 0010 незначащими являются два первых нуля.

В нашем случае все нули значащие, их количество 4.

Ответ: 4.

Перевод десятичных дробей из десятичной системы счисления в любую другую

Для перевода десятичной дроби из десятичной системы в любую другую следует умножать дробь, а затем дробные части произведений, на основание новой системы, пока дробная часть не станет равной 0 или до достижения указанной точности. Затем целые части выписать начиная с первой.

Пример 1.7

Переведите десятичное число 0,816 в двоичную систему с точностью до сотых.

Решение

<u>0</u> ,	816
	2
<u>1</u> ,	632
	2
<u>1</u> ,	264

$$0,816_{10} = 0,11_2.$$

Ответ: $0,11_2$.

Пример 1.8

Переведите число 0,8125 в восьмеричную систему.

Решение:

По умолчанию число 0,8125 считается десятичным.

<u>0</u>		8125
		8
<u>6</u>		5
		8
<u>4</u>		0

$$0,8125_{10} = 0,64_8.$$

Ответ: $0,64_8$.

Перевод смешанных чисел из десятичной системы счисления в любую другую

Если необходимо перевести смешанное число из десятичной системы в любую другую, следует перевести целую и дробную части, а затем записать, разделив десятичной запятой.

Пример 1.9

Сколько единиц в двоичной записи десятичного числа 14,125?

Решение:

Переведем целую часть числа в двоичную систему:

14		2	
14		7	2
<u>0</u>		6	3
		<u>1</u>	2
			<u>1</u>

Переведем дробную часть числа в двоичную систему:

<u>0,</u>	125
	2
<u>0,</u>	25
	2
<u>0,</u>	5
	2
<u>1,</u>	1

Соединим целую и дробную части:

$$14,125_{10} = 1110,001_2.$$

Количество единиц равно 4.

Ответ: 4.

Перевод чисел из двоичной системы в систему с основанием 2^n (степень двойки)

Для перевода числа из двоичной системы в систему с основанием 2^n следует разбить двоичную запись числа на группы по n цифр, начиная справа, при необходимости дополняя запись незначащими нулями. Каждую группу перевести в новую систему отдельно.

Пример 1.10

Двоичное число 110110 соответствует восьмеричному числу

- 1) 36 2) 54 3) 66 4) 72

Решение

Нам нужно перевести число из системы с основанием 2 в систему с основанием 8. $8=2^3$, значит, разделим двоичную запись на группы по 3 цифры, начиная справа:

$$110110_2.$$

Переведем отдельно каждую группу из трех цифр в восьмеричное число (пользуясь таблицей соответствия):

$$110_2 = 6_8;$$

$$110110_2 = 66_8.$$

Ответ: 3.

Перевод чисел системы с основанием 2^n (степень двойки) в двоичную

Для перевода числа из системы с основанием 2^n в двоичную следует каждую цифру исходной системы отдельно перевести в группу из n цифр двоичной, при необходимости дополняя запись незначащими нулями.

Пример 1.11

Как выглядит число BOC_{16} в двоичной системе счисления?

1) 110010001010_2

2) 101100001100_2

3) 101100010001_2

4) 101000011100_2

Решение

$16 = 2^4$. Значит, переведем каждую цифру исходного числа в 4-символьное число двоичной системы (используя таблицу соответствия):

$$B_{16} = 1011_2;$$

$$O_{16} = 0000_2;$$

$$C_{16} = 1100_2.$$

$$BOC_{16} = 101100001100_2.$$

Ответ: 2.

Пример 1.12

Укажите наименьшее четырёхзначное восьмеричное число, двоичная запись которого содержит 5 единиц. В ответе запишите только само восьмеричное число, основание системы счисления указывать не нужно.

Решение

Наименьшее четырехзначное восьмеричное число — это 1000_8 . Его двоичную запись получим, переведя отдельно каждую цифру числа в трехсимвольные числа (т.к. $8 = 2^3$). $1000_8 = 001\ 000\ 000\ 000_2 = 1\ 000\ 000\ 000_2$. Но нам нужно, чтобы двоичная запись содержала 5 единиц и число было наименьшим. Заменим в четырех младших разрядах 0 на 1, получим $1\ 000\ 001\ 111_2$. Это и есть искомое число. Осталось перевести его в восьмеричную систему: $1\ 000\ 001\ 111_2 = 1017_8$.

Ответ: 1017.

Задача № 16.

Позиционные системы счисления

Всю основную информацию, необходимую для решения этой задачи, мы уже знаем из п. 1.1. Осталось понять несколько несложных моментов.

Рассмотрим десятичное число 875. Последняя цифра числа (5) — это остаток от деления числа 875 на 10. Последние две цифры образуют число 75 — это остаток от деления числа 875 на 100. Аналогичные утверждения справедливы для любой системы счисления:

Последняя цифра числа — это остаток от деления этого числа на основание системы счисления.

Последние две цифры числа — это остаток от деления числа на основание системы счисления в квадрате.

Например, $212_3 = 2 + 1 \cdot 3 + 2 \cdot 3^2 = 23_{10}$. Разделим 23 на основание системы 3, получим 7 и 2 в остатке (2 — это последняя цифра числа в троичной системе). Разделим 23 на 9 (основание в квадрате), получим 18 и 5 в остатке ($5 = 12_3$).

Вернемся опять к привычной десятичной системе. Число $10^5 = 100000$. То есть 10 в степени k — это единица и k нулей.

Аналогичное утверждение справедливо для любой системы счисления:

Основание системы счисления в степени k в этой системе счисления записывается как единица и k нулей.

Например, $2^4 = 16_{10} = 10000_2$.

Поиск основания системы счисления

Пример 1.13

В системе счисления с некоторым основанием десятичное число 27 записывается в виде 30. Укажите это основание.

Решение

Обозначим искомое основание x .

Тогда $27 = 30_x = 0 \cdot x^0 + 3 \cdot x^1 = 3 \cdot x$. То есть $x = 9$.

Ответ: 9.

Пример 1.14

В системе счисления с некоторым основанием десятичное число 13 записывается в виде 111. Укажите это основание.

Решение

Обозначим искомое основание x .

Тогда $13 = 111_x = 1 \cdot x^0 + 1 \cdot x^1 + 1 \cdot x^2$;

$x^2 + x + 1 = 13$;

$$x^2 + x - 12 = 0.$$

Решаем квадратное уравнение, получаем корни 3 и -4 . Поскольку основание системы счисления не может быть отрицательным, ответ 3.

Ответ: 3.

Пример 1.15

Укажите через запятую в порядке возрастания все основания систем счисления, в которых запись числа 29 оканчивается на 5.

Решение

Если в некоторой системе число 29 оканчивается на 5, то уменьшенное на 5 число ($29 - 5 = 24$) оканчивается на 0. Ранее мы уже говорили, что число оканчивается на 0 в том случае, когда оно без остатка делится на основание системы. То есть нам нужно найти все такие числа, которые являются делителями числа 24. Эти числа: 2, 3, 4, 6, 8, 12, 24. Заметим, что в системах счисления с основанием 2, 3, 4 нет числа 5 (а в формулировке задачи число 29 оканчивается на 5), значит, остаются системы с основаниями: 6, 8, 12, 24.

Ответ: 6, 8, 12, 24.

Пример 1.16

Укажите через запятую в порядке возрастания все основания систем счисления, в которых запись числа 71 оканчивается на 13.

Решение

Обозначим основание системы через X .

Если в некоторой системе число оканчивается на 13, то основание этой системы не меньше 4 (иначе там нет цифры 3). То есть $X \geq 4$.

Поскольку число оканчивается на 13, мы можем представить его как $3 + X \cdot 1 + X^2 \cdot Z$, где Z — некоторое целое положительное число.

$$\text{Тогда } 71 = 3 + X \cdot 1 + X^2 \cdot Z$$

$$Z = \frac{71 - 3 - X \cdot 1}{X^2} = \frac{68 - X}{X^2}$$

Заметим, что самое большое значение Z достигается при самом маленьком X . То есть при $X = 4$. То есть

$$Z \leq \frac{68 - 4}{4^2}$$

$$Z \leq \frac{64}{16}$$

$$Z \leq 4$$

Подставим последовательно все значения Z от 0 до 4 в уравнение

$$71 = 3 + X \cdot 1 + X^2 \cdot Z$$

$Z = 0$	$Z = 1$	$Z = 2$
$71 = 3 + X$ $X = 68$	$71 = 3 + X \cdot 1 + X^2$ $X^2 + X - 68 = 0$ $D = 1 + 272 = 273$ Целочисленных корней нет	$71 = 3 + X \cdot 1 + 2 \cdot X^2$ $2 \cdot X^2 + X - 68 = 0$ $D = 1 + 544 = 545$ Целочисленных корней нет

$Z = 3$	$Z = 4$
$71 = 3 + X \cdot 1 + 3 \cdot X^2$ $3 \cdot X^2 + X - 68 = 0$ $D = 1 + 272 = 817$ Целочисленных корней нет	$71 = 3 + X \cdot 1 + 4 \cdot X^2$ $4 \cdot X^2 + X - 68 = 0$ $D = 1 + 1088 = 1089$ Целочисленный положительный корень $X = \frac{-1 + 33}{8} = 4$

Получили удовлетворяющие условию основания систем счисления 4 и 68.

Ответ: 4, 68.

Поиск чисел по условиям

Пример 1.17

Укажите через запятую в порядке возрастания все десятичные числа, не превосходящие 25, запись которых в системе счисления с основанием 4 оканчивается на 11.

Решение

Для начала выясним, как выглядит число 25 в системе счисления с основанием 4.

$25_{10} = 121_4$. То есть нам нужно найти все числа не больше 121_4 , запись которых оканчивается на 11. По правилу последовательного счета в системе с основанием 4, получаем числа 11_4 и 111_4 . Переводим их в десятичную систему счисления:

$$11_4 = 1 \cdot 4^0 + 1 \cdot 4^1 = 5_{10};$$

$$111_4 = 1 \cdot 4^0 + 1 \cdot 4^1 + 1 \cdot 4^2 = 21_{10}.$$

Ответ: 5, 21.

Решение уравнений

Пример 1.18

Решите уравнение: $121_x + 1_{10} = 101_7$

Ответ запишите в троичной системе (основание системы счисления в ответе писать не нужно).

Решение

$121_x + 1_{10} = 101_7$ Переведем все числа в десятичную систему счисления:

$$1 \cdot x^0 + 2 \cdot x^1 + 1 \cdot x^2 + 1 = 1 \cdot 7^0 + 0 \cdot 7^1 + 1 \cdot 7^2;$$

$$1 + 2 \cdot x + x^2 + 1 = 1 + 49;$$

$$x^2 + 2 \cdot x - 48 = 0.$$

Квадратное уравнение имеет корни -8 и 6 . $x = 6_{10}$ (так как основание системы не может быть отрицательным).
 $6_{10} = 20_3$.

Ответ: 20.

Подсчет количества нулей (единиц, двоек и т.д.) в записи значения выражения в системе счисления с основанием p

Для решения этого типа задач нам нужно вспомнить, как происходит сложение и вычитание «в столбик».

При сложении происходит поразрядное суммирование записанных друг под другом цифр, начиная с младших разрядов. В случае, если полученная сумма двух цифр больше или равна основанию системы счисления, под суммируемыми цифрами записывается остаток от деления этой суммы на основание системы, а целая часть от деления этой суммы на основание системы прибавляется к сумме следующих разрядов.

При вычитании происходит поразрядное вычитание записанных друг под другом цифр, начиная с младших разрядов. В случае, если первая цифра меньше второй, мы «занимаем» у соседнего (большого) разряда единицу. Занимаемая единица в текущем разряде равна основанию системы счисления. В привычной десятичной системе — это 10, в двоичной — 2, в троичной — 3 и т.д.

Пример 1.19

Сколько единиц содержится в двоичной записи значения выражения: $4^{2020} + 2^{2017} - 15$?

Решение

Представим все числа выражения как степени двойки:
 $4^{2020} + 2^{2017} - 15 = 2^{4040} + 2^{2017} - 2^4 + 2^0$.

В двоичной записи двойка в степени n выглядит как 1 и n нулей. Тогда, суммируя 2^{4040} и 2^{2017} , получим число, содержащее 2 единицы:

$$\begin{array}{r}
 \text{4040} \\
 \overbrace{100000 \dots 000 \dots 000000} \\
 + \quad \quad \quad \underbrace{100 \dots 000000}_{2017} \\
 \hline
 100000 \dots 100 \dots 000000 \\
 \quad \quad \quad \underbrace{\hspace{10em}}_{2017}
 \end{array}$$

Теперь вычтем из получившегося числа 10000. По правилам вычитания при недостатке занимаем у следующего разряда.

$$\begin{array}{r}
 100000 \dots 100 \dots 00000 \\
 \quad \quad \quad \underbrace{\hspace{10em}}_{2017} \\
 \quad \quad \quad 10000 \\
 \hline
 100000 \dots 0111 \dots 10000 \\
 \quad \quad \quad \underbrace{\hspace{5em}}_{2013} \quad \underbrace{\hspace{5em}}_4
 \end{array}$$

Теперь прибавляем к получившемуся числу 1:

$$\begin{array}{r}
 100000 \dots 0111 \dots 10000 \\
 + \quad \quad \quad \quad \quad \quad \quad 1 \\
 \hline
 \underbrace{100000}_{1} \dots \underbrace{0111 \dots 10001}_{2013} \underbrace{1}_{1}
 \end{array}$$

Видим, что у результата $2013 + 1 + 1 = 2015$ единиц.

Ответ: 2015.

Пример 1.20

Значение арифметического выражения $9^8 + 3^5 - 9$ записали в системе счисления с основанием 3. Сколько цифр «2» содержится в этой записи?

Решение

Представим все числа выражения как степени тройки:
 $3^{16} + 3^5 - 3^2$.

В системе счисления с основанием три число 3^n выглядит как 1 и n нулей. Тогда, суммируя 3^{16} и 3^5 , получим 17-разрядное число, содержащее 2 единицы (в первой и шестой справа позициях) и 15 нулей:

$$\begin{aligned} &10000000000000000 + \\ &+ 00000000000100000 = \\ &= 100000000000100000. \end{aligned}$$

Вычтем из полученного числа 3^2 (это единица и 2 нуля):

$$100000000000100000 - 100$$

По правилам вычитания при недостатке занимаем у следующего разряда:

$$100000000000022200.$$

Ответ: 3.

Задача № 3

Анализ информационных моделей

В своей деятельности человек повсеместно использует модели, т.е. создает образ, упрощенную копию того объекта, с которым ему приходится иметь дело.

Модель — это искусственно созданный объект, дающий упрощенное представление о реальном объекте, процессе или явлении, отражающий существенные стороны изучаемого объекта с точки зрения цели моделирования.

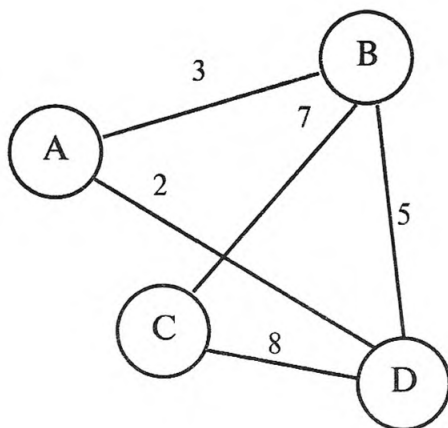
Моделирование — это построение моделей, предназначенных для изучения объектов, процессов или явлений.

Распространенными информационными моделями являются графики, схемы, таблицы, диаграммы. Одним из распространенных видов моделей являются графы.

Граф — это один из способов графического представления информации. Объекты представлены в нем как вершины (узлы), а связи между объектами как ребра (дуги). То есть граф — это набор вершин и связывающих их ребер.

Путь в графе — это конечная последовательность вершин, каждая из которых (кроме последней) соединена со следующей ребром. Граф может содержать циклы (первая вершина пути может совпадать с последней).

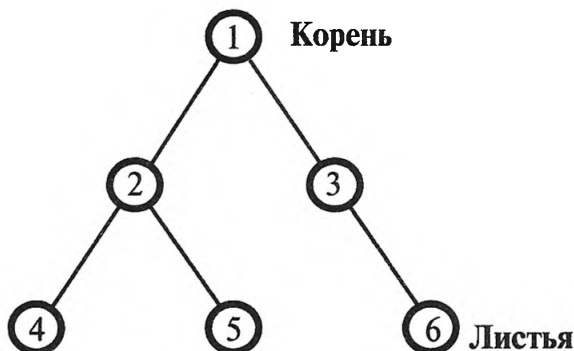
Обычно в задачах используют взвешенный граф, т.е. граф, в котором с каждым ребром связано число (вес). Например, расстояние, стоимость и т.д.



Граф может задаваться таблицей, в которой на пересечении строки и столбца с наименованиями вершин записано числовое значение (вес) ребра, соединяющего эти вершины.

	A	B	C	D
A		3		2
B	3		7	5
C		7		8
D	2	5	8	

Дерево — это граф, не имеющий циклов. В дереве существует один-единственный путь между любой парой вершин. Одна из вершин дерева (корень) не имеет входящих ребер, все остальные имеют ровно одно входящее ребро. Вершины, у которых нет исходящих ребер, называются листьями.

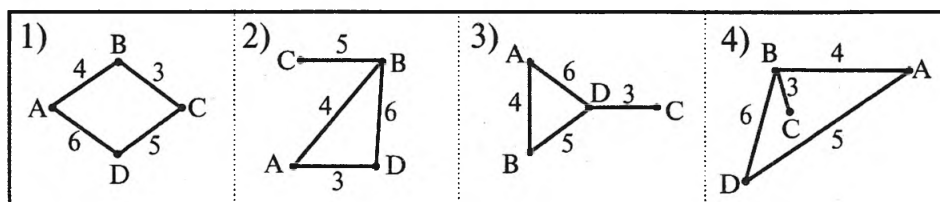


Поиск графа, соответствующего таблице

Пример 2.1

В таблице приведена стоимость перевозок между соседними железнодорожными станциями. Укажите схему, соответствующую таблице.

	A	B	C	D
A		4		5
B	4		3	6
C		3		
D	5	6		



Решение

Сравним значения таблицы и схем:

Согласно таблице вершина *A* должна быть связана с вершинами *B* (значение 4) и *D* (значение 5). То есть $AB = 4$, $AD = 5$. На схеме значения указаны около соответствующего ребра. Сразу отбрасываем 1), 2), 3) схемы, так как на них AD не равно 5. Для уверенности проверим все остальные ребра схемы 4): $BC = 3$, $BD = 6$, что совпадает со значениями таблицы. Правильная схема 4).

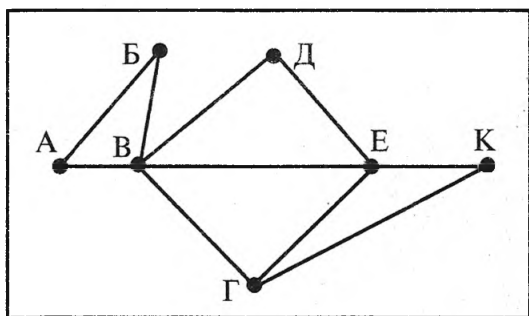
Ответ: 4

Анализ информации в таблице и графе

Пример 2.2

На рисунке справа схема дорог Н-ского района изображена в виде графа, в таблице содержатся сведения о длинах этих дорог (в километрах).

	П1	П2	П3	П4	П5	П6	П7
П1		45		10			
П2	45			40		35	
П3						60	
П4	10	40				20	35
П5			15			55	
П6		55	60	20	55		45
П7				35		45	



Так как таблицу и схему рисовали независимо друг от друга, то нумерация населенных пунктов в таблице никак не связана с буквенными обозначениями на графе. Определите, какова длина дороги из пункта В в пункт Е. В ответе запишите целое число — так, как оно указано в таблице.

Решение

На графе из вершины В выходит 5 ребер, значит, в таблице соответствующий пункт должен иметь дороги в 5 других (строка должна содержать 5 заполненных клеток). Такой пункт в таблице один: П6.

На графе из вершины Е выходит 4 ребра, значит, в таблице соответствующий пункт должен иметь дороги в 4 других (строка должна содержать 4 заполненные клетки). Такой пункт в таблице один: П4.

Таким образом, нам нужно найти расстояние между П6 и П4. Согласно таблице оно равно 20.

Ответ: 20.

Поиск информации в таблице по условию

Пример 2.3

Между четырьмя местными аэропортами: ЛУГОВОЕ, ДЯТЛОВО, НИКИТИНО и ОРЕХОВО, ежедневно выполняются авиарейсы. Приведен фрагмент расписания перелетов между ними:

Аэропорт вылета	Аэропорт прилета	Время вылета	Время прилета
ДЯТЛОВО	ЛУГОВОЕ	10:15	10:55
ЛУГОВОЕ	НИКИТИНО	10:20	11:00
ОРЕХОВО	ЛУГОВОЕ	10:25	12:05
ЛУГОВОЕ	ДЯТЛОВО	10:30	11:15
НИКИТИНО	ЛУГОВОЕ	10:55	11:40
ОРЕХОВО	ДЯТЛОВО	11:10	11:55
ЛУГОВОЕ	ОРЕХОВО	11:50	13:30
ДЯТЛОВО	ОРЕХОВО	12:00	12:50
НИКИТИНО	ОРЕХОВО	12:05	12:55
ОРЕХОВО	НИКИТИНО	12:10	12:55

Путешественник оказался в аэропорту ЛУГОВОЕ в полночь. Определите самое раннее время, когда он может попасть в аэропорт ОРЕХОВО. Считается, что путешественник успевает совершить пересадку в аэропорту, если между временем прилета в этот аэропорт и временем вылета проходит не менее часа.

- 1) 12:05 2) 12:50 3) 12:55 4) 13:30

Решение

Можно, конечно, решить эту задачу, просто глядя на таблицу и перебирая подходящие варианты, но есть риск

ошибиться или пропустить нужную строчку. Поэтому рекомендую нарисовать дерево всех возможных путей из аэропорта ЛУГОВОЕ в ОРЕХОВО:



Средняя ветка не подходит, так как между прилетом в аэропорт ДЯТЛОВО (11:15) и вылетом из ДЯТЛОВО в ОРЕХОВО (12:00) интервал меньше часа.

Из оставшихся двух выбираем раннее время прилета: 12:55.

Ответ: 3.

Выбор таблицы по условию

Пример 2.4

В таблицах приведена протяженность автомагистралей между соседними населенными пунктами. Если пересечение строки и столбца пусто, то соответствующие населенные пункты не являются соседними. Укажите номер таблицы, для которой выполняется условие «Максимальная протяженность маршрута от пункта С до пункта В не больше 6». Протяженность маршрута складывается из протяженности автомагистралей между соответствующими соседними населенными пунктами. При этом через любой населенный пункт маршрут должен проходить не более одного раза.

1)

	A	B	C	D	E
A		4	3		7
B	4			2	
C	3			6	
D		2	6		1
E	7			1	

2)

	A	B	C	D	E
A		2	5		6
B	2			3	
C	5				
D		3			1
E	6			1	

3)

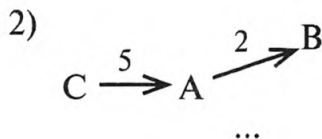
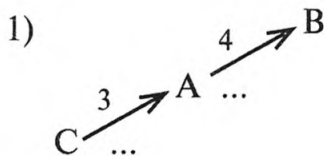
	A	B	C	D	E
A			2	2	6
B				2	
C	2			2	
D	2	2	2		
E	6				

4)

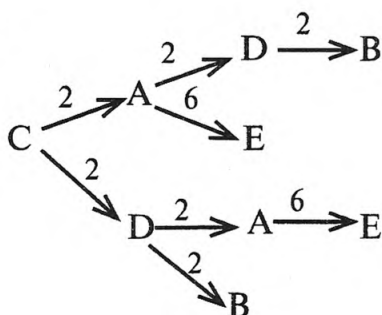
	A	B	C	D	E
A		5	2		6
B	5			5	
C	2			2	
D		5	2		3
E	6			3	

Решение

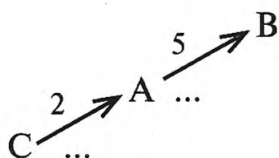
По каждой из схем построим дерево с корнем в точке С и листьями в точке В. При этом нам не нужно строить дерево полностью. Как только найдена ветка с протяженностью больше 6, делаем вывод, что таблица не удовлетворяет указанному условию:



3)



4)



Таблицы 1), 2) и 4) отвергаем уже при анализе первой ветки дерева.

В таблице 3) две ветки вообще не приведут в В, а две другие имеют суммарную длину, не превышающую 6.

Ответ: 3.

Поиск кратчайшего пути по таблице

Пример 2.5

Между населенными пунктами A, B, C, D, E, F, Z построены дороги, протяжённость которых приведена в таблице. (Отсутствие числа в таблице означает, что прямой дороги между пунктами нет.)

	A	B	C	D	E	F	Z
A		4	6				33
B	4		1				
C	6	1		2	10		
D			2		4		
E			10	4		3	8
F					3		2
Z	33				8	2	

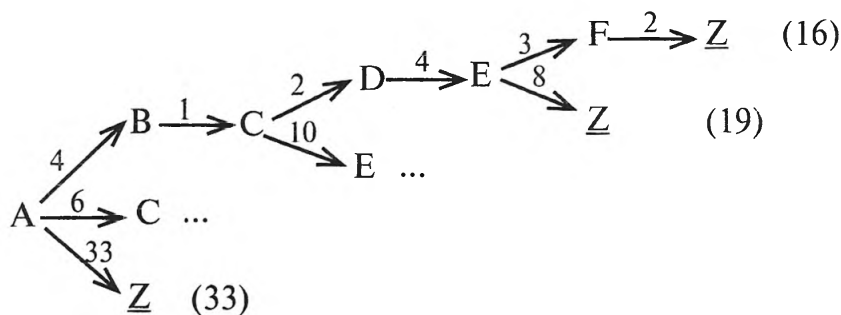
Определите длину кратчайшего пути между пунктами A и Z (при условии, что передвигаться можно только по построенным дорогам).

- 1) 13 2) 16 3) 19 4) 21

Решение

При решении этой задачи тоже не следует полагаться на простой визуальный анализ таблицы. Чтобы избежать ошибок, построим дерево с корнем в вершине A и листьями в вершине Z . При этом нам не нужно выписывать все ветки. Второй путь из A в C ($AC = 6$) длиннее первого ($ABC = 5$), значит и весь маршрут через него будет длиннее.

Второй путь из C в E ($CE = 10$) длиннее первого ($CDE = 6$), значит и весь маршрут через него будет длиннее.



Нам остается сложить длины всех отрезков и выбрать маршрут с наименьшей длиной. Это верхняя ветка дерева с длиной 16.

Ответ: 2.

Пример 2.6

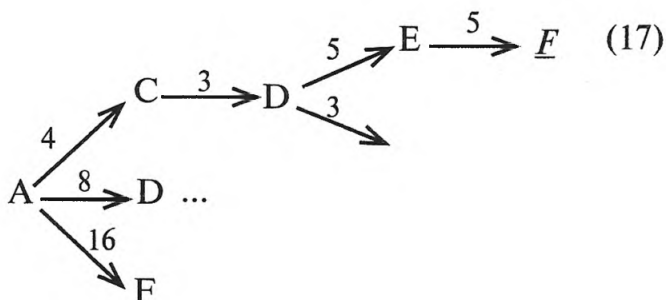
Между населенными пунктами A, B, C, D, E, F построены дороги, протяжённость которых приведена в таблице. (Отсутствие числа в таблице означает, что прямой дороги между пунктами нет.)

	A	B	C	D	E	F
A		2	4	8		16
B	2			3		
C	4			3		
D	8	3	3		5	3
E				5		5
F	16			3	5	

Определите длину кратчайшего пути между пунктами A и F , проходящего через пункт E и не проходящего через пункт B . Передвигаться можно только по указанным дорогам.

Решение

Построим дерево с корнем в вершине A и листьями в вершине F . При этом сразу опустим ветки, проходящие через вершину B :



Второй путь в D ($AD = 8$) длиннее первого ($ACD = 7$), значит, и весь маршрут через него будет длиннее. Не будем его рассматривать.

Из оставшихся веток только верхняя проходит через вершину E . Длина пути равна 17.

Ответ: 17.

Задача № 15

Поиск путей в графе

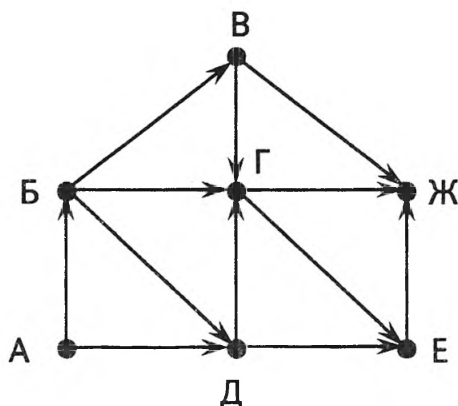
В этой задаче мы имеем дело с ориентированным графом (графом, у которого ребра имеют направление). То есть ребра имеют вид стрелок. Две вершины, соединенные направленной стрелкой, называются смежными. Вершина, из которой выходит стрелка, называется предком, а вершина, в которую входит стрелка, — потомком.

Несложно понять, что количество путей, которыми можно попасть в некоторую вершину, равно сумме количеств путей предков этой вершины.

Подсчет путей в ориентированном графе

Пример 2.7

На рисунке — схема дорог, связывающих города А, Б, В, Г, Д, Е, Ж, З, И, К. По каждой дороге можно двигаться только в одном направлении, указанном стрелкой. Сколько существует различных путей из города А в город Ж?

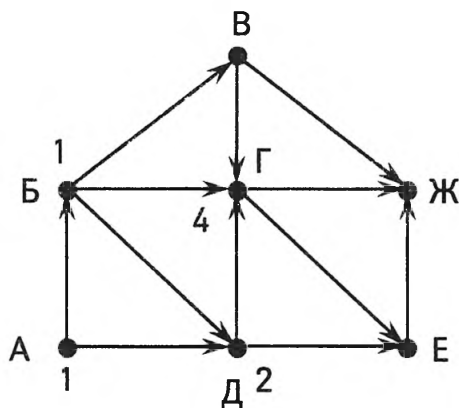


Решение

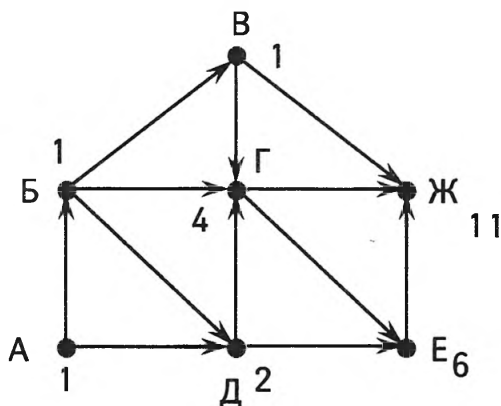
Каждой вершине, начиная с начальной (А), поставим в соответствие индекс, равный количеству путей, которыми

можно попасть в эту вершину. Для вершины А (начало пути) индекс всегда равен 1 (в начало пути можно попасть единственным образом — никуда не двигаясь). Теперь сформулируем правило: индекс вершины равен сумме индексов его предков. Исходя из этого индекс Б равен 1 (предок у Б один — вершина А).

У вершины Д предками являются А и Б, значит, индекс вершины Д равен $1 + 1 = 2$.



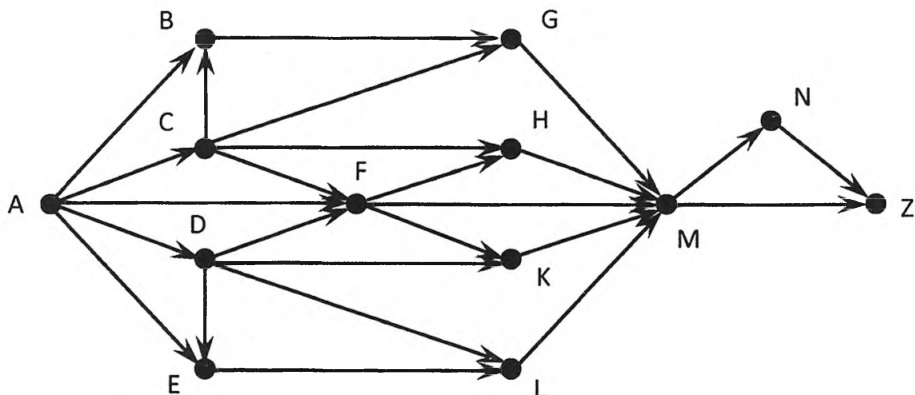
Очевидно, что мы можем посчитать индекс только тех вершин, индексы предков которых уже посчитаны. Например, мы не можем посчитать индекс Г, пока не посчитан индекс В. Двигаясь последовательно, мы рассчитаем индексы всех вершин. Индекс вершины Ж и будет ответом задачи.



Ответ: 11.

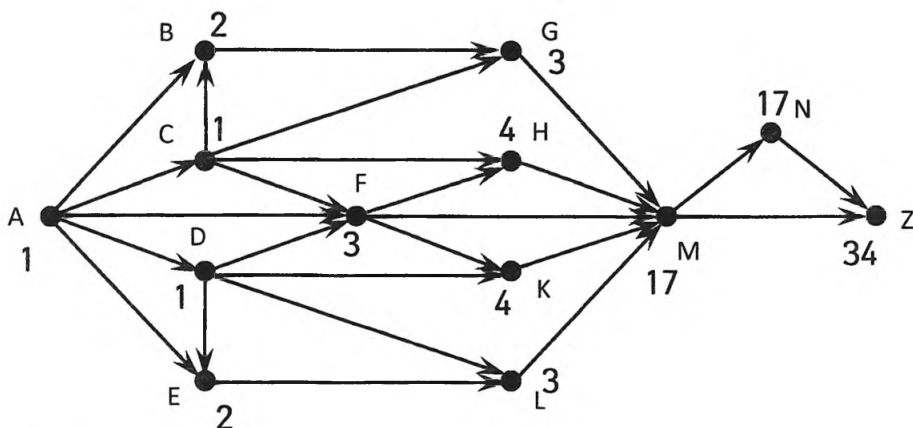
Пример 2.8

На рисунке — схема дорог, связывающих города $A, B, C, D, E, F, G, H, K, L, M, N, Z$. По каждой дороге можно двигаться только в одном направлении, указанном стрелкой. Сколько существует различных путей из города A в город Z ?



Решение:

Задача решается абсолютно так же, как предыдущая. Присваиваем начальной вершине A индекс 1 и последовательно считаем индексы всех вершин до Z :



Ответ: 34.

Задача № 4

Файловая система. Базы данных

Порядок, определяющий способ организации, хранения и именования данных на носителях информации в компьютерах, называется файловой системой.

Файл (от английского «папка») — это некоторая совокупность данных, хранимая и обрабатываемая как единое целое.

Файл имеет имя и расширение. Расширение указывает на то, какой именно тип данных находится в файле (программа, текст, рисунок и т.д.). Расширение отделено от имени файла точкой (например, test.doc).

Примеры некоторых типов файлов:

Исполняемые файлы — расширения .exe, .com;

Текстовые файлы — расширения .txt, .doc;

Архивные файлы — расширения .rar, .zip;

Звуковые файлы — расширения .mp3, .wav;

Файлы, содержащие изображения — .jpg, .gif, .bmp.

И т.д.

В ОС *Windows* и других операционных системах используется типовая древовидная структура файловой системы.

Корневой каталог обозначается буквой — именем логического диска. После буквы ставится двоеточие и обратный слэш: C:\.

Полный путь к файлу состоит из имени корневого каталога и имен последовательно вложенных друг в друга каталога

логов, разделенных обратным слэшем. В конце пути находится сам файл: C:\Study\Subject\Math\text1.txt.

Для выделения, фильтрации группы файлов используют маску (шаблон) имени файла.

Для задания масок кроме обычных символов, допустимых в именах файлов, используют специальные символы:

«*» — заменяет любое количество (в том числе и нулевое) любых символов;

«?» — заменяет один и только один обязательно стоящий в указанном месте символ.

Например, по маске «*.» будут отобраны вообще все файлы, по маске «*.txt» — файлы с расширением .txt, по маске «as?.doc» — файлы, с расширением .doc, имена которых начинаются на «as» и имеют обязательный непустой третий символ.

Если установлена сортировка по имени или типу, сравнение идет по кодам символов. При этом если задана сортировка, к примеру, по имени, то при наличии одинаковых имен сортировка будет применена к расширению.

Для хранения и анализа большого объема информации организуют Базы Данных. Под Базой Данных понимают организованную в соответствии с некоторыми правилами, структурированную совокупность логически связанных данных. Эти данные предназначены для удобного совместного хранения и анализа.

Реляционная База Данных состоит из связанных между собой таблиц.

Перемещение по каталогам

Пример 3.1

Учитель работал в каталоге C:\Учебный материал\11 класс\Контрольные работы.

Затем перешел в дерево каталогов на уровень выше, спустился в подкаталог Теоретический материал и

удалил из него файл **Введение**. Каково полное имя файла, который удалил преподаватель?

- 1) **C:\Учебный материал\11 класс\Введение.doc**
- 2) **C:\Учебный материал\11 класс\Теоретический материал\Введение.doc**
- 3) **C:\Учебный материал\Теоретический материал\Введение.doc**
- 4) **C:\Учебный материал\Введение\Теоретический материал.doc**

Решение

Последовательно повторим действия учителя. После того, как он перешел в дереве каталогов на уровень выше, он оказался в каталоге **C:\Учебный материал\11 класс**, спустившись в подкаталог **Теоретический материал**, он оказался в каталоге **C:\Учебный материал\11 класс\Теоретический материал**. Именно там учитель и удалил файл **Введение**. Значит полное имя удаленного файла **C:\Учебный материал\11 класс\Теоретический материал\Введение.doc**.

Ответ: 2

Пример 3.2

Находясь в корневом каталоге только что отформатированного диска, ученик создал 3 каталога. Затем в каждом из них он создал еще по 4 каталога. Сколько всего каталогов оказалось на диске, включая корневой?

- 1) 12 2) 13 3) 15 4) 16

Решение

Обратим внимание на словосочетание «включая корневой». Значит, на диске существует корневой каталог, 3 вновь созданных каталога и в каждом из трех еще по 4: $1 + 3 + 3 \cdot 4 = 16$.

Ответ: 4.

Отбор файла по маске

Пример 3.3

Для групповых операций с файлами используются маски имен файлов. Маска представляет собой последовательность букв, цифр и прочих допустимых в именах файлов символов, в которых также могут встречаться следующие символы: Символ «?» (вопросительный знак) означает ровно один произвольный символ. Символ «*» (звездочка) означает любую последовательность символов произвольной длины, в том числе «*» может задавать и пустую последовательность. Определите, какое из указанных имен файлов удовлетворяет маске: **?vi*r.?xt**

- 1) vir.txt 2) ovir.txt 3) ovir.xt 4) virr.txt

Решение

Последовательно переберем все файлы: 1) не подходит, т.к. «?» в первой позиции маски означает обязательное наличие непустого символа перед буквой v, 2) полностью удовлетворяет условию маски, 3) не подходит, т.к. «?» после точки означает обязательное наличие непустого символа перед буквой x, 4) не подходит по той же причине, что и 1).

Ответ: 2.

Пример 3.4

Для групповых операций с файлами используются маски имен файлов. Маска представляет собой последовательность букв, цифр и прочих допустимых в именах файлов символов, в которых также могут встречаться следующие символы: Символ «?» (вопросительный знак) означает ровно один произвольный символ. Символ «*» (звездочка) означает любую последовательность символов произвольной длины, в том числе «*»

может задавать и пустую последовательность. Определите, какое из указанных имен файлов не удовлетворяет маске: sys??.*

- 1) syste.m 2) sys23.exe 3) system.dll 4) syszx.problem

Решение

По условиям маски файл должен иметь любое расширение, а имя файла должно начинаться на sys и оканчиваться ровно двумя непустыми символами. Этому условию не удовлетворяет только файл 3). Он содержит в имени не два, а три символа после sys.

Ответ: 3.

Поиск маски по набору файлов

Пример 3.5

Для групповых операций с файлами используются маски имён файлов. Маска представляет собой последовательность букв, цифр и прочих допустимых в именах файлов символов, в которых также могут встречаться следующие символы:

Символ «?» (вопросительный знак) означает ровно один произвольный символ.

Символ «*» (звездочка) означает любую последовательность символов произвольной длины, в том числе «*» может задавать и пустую последовательность.

Определите, по какой из масок может быть выбрана указанная группа файлов:

comfort.c
cobalt.cpp
coat.c
cost.cpp

- 1) co?t.c?
2) c*.c*

3) co*t.c??

4) c*.???

Решение

Последовательно применим каждую маску к набору файлов:

Маска 1) требует, чтобы имя состояло ровно из 4-х символов, под эти требования не подходят файлы comfort.c и cobalt.cpp.

Маска 2) требует только, чтобы имя и расширение файла начинались на с. Под эти требования подходят все файлы.

Маска 3) требует, чтобы расширение содержало 3 символа, под эти требования не подходят файлы comfort.c и coat.c.

Маска 4) тоже требует, чтобы расширение содержало 3 символа. Аналогично предыдущему случаю не подходят файлы comfort.c и coat.c.

Ответ: 2.

Пример 3.6

Для групповых операций с файлами используются маски имён файлов. Маска представляет собой последовательность букв, цифр и прочих допустимых в именах файлов символов, в которых также могут встречаться следующие символы:

Символ «?» (вопросительный знак) означает ровно один произвольный символ.

Символ «*» (звездочка) означает любую последовательность символов произвольной длины, в том числе «*» может задавать и пустую последовательность.

В каталоге находится 6 файлов:

chifera.dat

chifera.doc

ferrum.doc

deLafer.doc

oferta.doc
tokoferol.docx

Определите, по какой из масок из каталога будет отобрано ровно 2 файла.

- 1) *fer?*.d*
- 2) ?*fer*.doc
- 3) *?fer*?.doc*
- 4) ?*fer?*.doc

Решение

Последовательно применим каждую маску к набору файлов:

По маске 1) будут отобраны файлы с расширением, начинающимся на d и именем, обязательно содержащим последовательность символов fer и еще один символ после. Под эту маску подходят все файлы, кроме deLafer.doc (5 файлов).

По маске 2) будут отобраны все файлы с расширением .doc и именем, обязательно содержащим последовательность символов fer и еще один символ до. Под эту маску подходят файлы: chifera.doc, deLafer.doc, oferta.doc (3 файла).

По маске 3) будут отобраны все файлы с расширением, начинающимся на .doc и именем, обязательно содержащим последовательность символов fer, один символ до этой последовательности и еще один символ после. Под эту маску подходят файлы: chifera.doc, oferta.doc, tokoferol.docx (3 файла).

По маске 4) будут отобраны все файлы с расширением .doc и именем, обязательно содержащим последовательность символов fer, один символ до этой последовательности и еще один символ после. Под эту маску подходят файлы: chifera.doc, oferta.doc, (2 файла).

Ответ: 4.

Пример 3.7

Для групповых операций с файлами используются маски имён файлов. Маска представляет собой последовательность букв, цифр и прочих допустимых в именах файлов символов, в которых также могут встречаться следующие символы:

Символ «?» (вопросительный знак) означает ровно один произвольный символ.

Символ «*» (звездочка) означает любую последовательность символов произвольной длины, в том числе «*» может задавать и пустую последовательность.

В каталоге находится 6 файлов:

maveric.map

maveric.mp3

taverna.mp4

revolver.mp4

vera.mp3

zveri.mp3

Ниже представлено восемь масок. Сколько из них таких, которым соответствуют ровно четыре файла из данного каталога?

ver.mp*	*?ver?*mp?	?*ver*.mp?*	*v*r*?.m?p*
???*???mp*	???*???m*	*a*.a*	*a*.p*

Решение

Последовательно применим каждую маску к набору файлов:

По маске *ver*.mp* будут отобраны файлы: maveric.mp3, taverna.mp4, revolver.mp4, vera.mp3, zveri.mp3 (5 файлов).

По маске *v*r*?.m?p* будут отобраны файлы: maveric.mp3, taverna.mp4, zveri.mp3 (3 файла).

По маске `?*ver*.mp?` будут отобраны файлы: `maveric.mp3`, `taverna.mp4`, `revolver.mp4`, `zveri.mp3` (4 файла).

По маске `*v*g*?.m?p*` будут отобраны файлы: `maveric.map` (1 файл).

По маске `???*???.mp*` будут отобраны файлы: `maveric.mp3`, `taverna.mp4`, `revolver.mp4` (3 файла).

По маске `???*???.m*` будут отобраны файлы: `maveric.map`, `maveric.mp3`, `taverna.mp4`, `revolver.mp4` (4 файла).

По маске `*a*.a*` будут отобраны файлы: `maveric.map` (1 файл).

По маске `*a*.r*` будут отобраны файлы: `maveric.map`, `maveric.mp3`, `taverna.mp4`, `vera.mp3` (4 файла).

Ответ: 3.

Сортировка файлов

Пример 3.8

Каталог содержит файлы с именами

- | | |
|-------------------------|-------------------------|
| а) <code>p5.pas</code> | г) <code>pq.p</code> |
| б) <code>p4.ppt</code> | д) <code>pq.pas</code> |
| в) <code>p12.pas</code> | е) <code>p12.ppt</code> |

Определите, в каком порядке будут показаны файлы, если выбрана сортировка по типу (по возрастанию).

- | | |
|------------------------|------------------------|
| 1) <code>вадгеб</code> | 2) <code>гавдбе</code> |
| 3) <code>вадгбе</code> | 4) <code>гвадеб</code> |

Решение

Отсортируем файлы по типу (по возрастанию). Так как все расширения начинаются на одну букву `p`, анализируем вторую букву расширения. На первом месте будет стоять файл, в расширении которого вообще нет второй буквы:

`pq.p` (г)

Затем идет файл с расширением `.pas`, но поскольку таких файлов несколько, анализируем имена файлов. Учитываем, что цифра «меньше» буквы.

p12.pas (в)

p5.pas (а)

pq.pas (д)

Дальше идут файлы с расширением .ppt.

p12.ppt (е)

p4.ppt (б)

Ответ: 4.

Базы данных. Анализ одной таблицы

Пример 3.9

На городской тур олимпиады по математике отбираются те учащиеся, кто набрал на районном туре не менее 10 баллов или полностью решил хотя бы одну из двух самых сложных задач (№ 6 или № 7). За полное решение задач 1–4 даётся 2 балла; задач 5, 6 — 3 балла; задачи 7 — 4 балла. Дан фрагмент таблицы результатов районного тура.

Фамилия	Пол	Задача № 1	Задача № 2	Задача № 3	Задача № 4	Задача № 5	Задача № 6	Задача № 7
Айвазян	ж	1	0	2	1	0	1	3
Житомирский	м	2	2	2	2	2	1	3
Иваненко	ж	2	1	1	0	1	2	3
Лимонов	м	1	1	1	1	1	2	3
Петраков	м	2	0	0	1	0	3	0
Рахимов	м	2	2	2	0	3	0	1
Суликашвили	ж	1	1	0	0	0	3	2
Толкачёва	ж	2	1	0	0	3	0	3

Сколько девочек из этой таблицы прошли на городской тур?

1) 1

2) 2

3) 3

4) 4

Решение

Отметим в таблице девочек (пол ж), у которых сумма баллов не менее 10, или задача № 6 решена на 3 балла, или задача № 7 решена на 4 балла.

Фамилия	Пол	Задача № 1	Задача № 2	Задача № 3	Задача № 4	Задача № 5	Задача № 6	Задача № 7
Айвазян	ж	1	0	2	1	0	1	3
Житомирский	м	2	2	2	2	2	1	3
Иваненко	ж	2	1	1	0	1	2	3
Лимонов	м	1	1	1	1	1	2	3
Петраков	м	2	0	0	1	0	3	0
Рахимов	м	2	2	2	0	3	0	1
Суликашвили	ж	1	1	0	0	0	3	2
Толкачёва	ж	2	1	0	0	3	0	3

Ответ: 2.

Базы данных. Анализ связанных таблиц

Пример 3.10

В фрагменте базы данных представлены сведения о родственных отношениях. На основании приведённых данных определите, сколько прямых потомков (т.е. детей и внуков) Павленко А.К. упомянуты в таблице 1.

Решение

Две таблицы связаны через поле ID.

По таблице 1 определим ID Павленко А.К.: 2155.

По таблице 2 определим ID детей Павленко А.К.: 2302, 3002.

По таблице 2 определим ID внуков (дети детей) Павленко А.К.: у 2302 дети 2431, 2511, 3193, а у 3002 дети 2586, 2570.

Таблица 1		
ID	Фамилия И.О.	Пол
2146	Кривич Л.П.	Ж
2155	Павленко А.К.	М
2431	Хитрук П.А.	М
2480	Кривич А.А.	М
2302	Павленко Е.А.	Ж
2500	Сокол Н.А.	Ж
3002	Павленко И.А.	М
2523	Павленко Т.Х.	Ж
2529	Хитрук А.П.	М
2570	Павленко П.И.	М
2586	Павленко Т.И.	Ж
2933	Симонян А.А.	Ж
2511	Сокол В.А.	Ж
3193	Биба С.А.	Ж
...	...	...

Таблица 2	
ID Родителя	ID Ребёнка
2146	2302
2146	3002
2155	2302
2155	3002
2302	2431
2302	2511
2302	3193
3002	2586
3002	2570
2523	2586
2523	2570
2529	2431
2529	2511
2529	3193
...	...

То есть общее число детей и внуков 7.

Ответ: 7.

Пример 3.11

Ниже приведены фрагменты таблиц базы данных учеников школы:

Код класса	Класс	Фамилия	Код класса
1	9-А	Букин	3
2	9-Б	Веселова	5
3	9-В	Викторов	8
4	10-А	Дятлов	3
5	10-Б	Егорова	2
6	10-В	Ершова	8
7	11-А	Жилин	5
8	11-Б	Захаров	2
9	11-В	Зуева	3

В каком классе учится наибольшее число учеников?

- 1) 11-Б 2) 9-В 3) 9-А 4) 10-Б

Решение

Две таблицы связаны через поле Код класса.

По таблице учеников видим, что наибольшее число учеников (3 человека) учится в классе с кодом 3.

По таблице классов находим, что коду 3 соответствует класс 9-В.

Ответ: 2.

Задача № 17

Запросы для поисковых систем

Для быстрого поиска информации в Интернете используют поисковые запросы. Поисковый запрос — это набор ключевых слов, соединенных знаками логических операций И, ИЛИ, НЕ.

Приоритет выполнения операций, если нет специально поставленных скобок, следующий: сначала НЕ, затем И, затем ИЛИ.

Нужно понимать, что операция И (одновременное выполнение условий) сокращает объем получаемого результата, а операция ИЛИ (выполнение хотя бы одного из условий), наоборот, увеличивает объем.

Если в запросе стоит фраза в кавычках, система будет искать точно такую фразу целиком.

Расположение запросов по возрастанию (убыванию)

Операция «И» (&) обозначает одновременное присутствие ключевых слов в искомых документах, а потому

уменьшает количество найденной информации. Чем больше ключевых слов соединены операцией «И», тем меньше количество найденной информации. И наоборот, операция «ИЛИ» (|) обозначает присутствие хотя бы одного ключевого слова в искомых документах, а потому увеличивает количество найденной информации.

Пример 3.12

В таблице приведены запросы к поисковому серверу. Расположите обозначения запросов в порядке возрастания количества страниц, которые найдет поисковый сервер по каждому запросу.

- А) реферат | математика | Гаусс
- Б) реферат | математика | Гаусс | метод
- В) реферат | математика
- Г) реферат & математика & Гаусс

Решение

Самое маленькое количество страниц будет отобрано по запросу с наибольшим количеством операций «И» (запрос Г), Самое большое количество страниц будет отобрано по запросу с наибольшим количеством операций «ИЛИ» (запрос Б). По запросу А будет отобрано больше страниц, чем по запросу В, т.к. запрос А содержит больше ключевых слов, связанных операцией «ИЛИ».

Ответ: ГВАБ.

Пример 3.13

В таблице приведены запросы к поисковому серверу, условно обозначенные буквами от А до Г. Расположите запросы в порядке возрастания количества страниц, которые найдет поисковый сервер по каждому запросу. Ответ запишите в виде последовательности соответствующих букв.

- А) (огурцы & помидоры) & (прополка | поливка)
- Б) огурцы | помидоры
- В) огурцы
- Г) огурцы & помидоры

Решение

Самое маленькое количество страниц будет отобрано по запросу А (на страницах должны упоминаться и огурцы, и помидоры, и прополка или поливка).

Следующий по количеству найденных страниц запрос Г (на страницах должны упоминаться и огурцы, и помидоры).

Затем идет запрос В (упоминаются только огурцы).

Самое большое количество страниц будет отобрано по запросу Б (упоминаются или огурцы или помидоры).

Ответ: АГВБ.

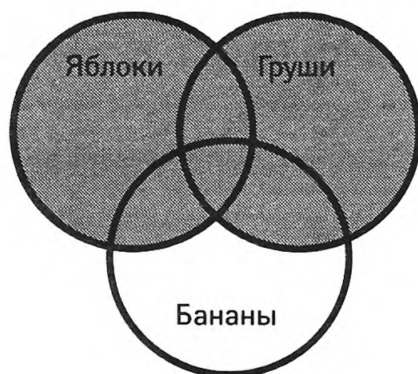
Подсчет найденных по запросу страниц

Принцип отбора информации по поисковым запросам хорошо иллюстрирует диаграмма Эйлера–Венна (круги Эйлера). На диаграмме множества изображаются пересекающимися кругами. Операция «И» (&) — это пересечение кругов, а операция «ИЛИ» (|) — это объединение кругов.

Например, обозначим кругами множества *Яблоки*, *Груши*, *Бананы*. По запросу *Яблоки & Груши & Бананы* будет отобрано пересечение (общая часть) всех трех кругов:



По запросу *Яблоки | Груши* будет отображено объединение двух кругов:



Пример 3.14

В таблице приведены запросы и количество страниц, которые нашел поисковый сервер по этим запросам в некотором сегменте Интернета:

Запрос	Количество страниц (тыс.)
шахматы теннис	7770
теннис	5500
шахматы & теннис	1000

Сколько страниц (в тысячах) будет найдено по запросу *шахматы*?

Решение

Нарисуем диаграмму Эйлера–Венна. Прием решения задачи состоит в подсчете количества страниц, соответствующее каждой области, ограниченной линиями:

Запросу *шахматы & теннис* соответствует средняя область (1000 тыс. страниц), а запросу *теннис* — весь правый круг (5500 тыс. страниц).

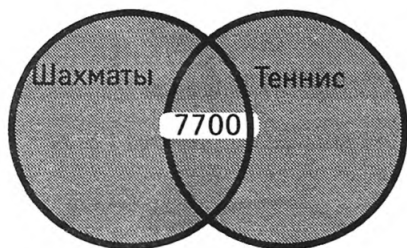
3. Технологии поиска и хранения информации



Тогда правый «обрезанный круг» — это $5500 - 1000 = 4500$:



Запросу *шахматы* | *теннис* соответствуют оба круга 7770), тогда левый «обрезанный круг» — это $7770 - 5500 = 2270$



Итак, мы посчитали количество страниц для каждой ограниченной линиями области:



Несложно увидеть, что по запросу *шахматы* будет найдено $2270 + 1000 = 3270$ тыс. страниц.

Ответ: 3270.

Пример 3.15

В таблице приведены запросы и количество страниц, которые нашел поисковый сервер по этим запросам в некотором сегменте Интернета:

Запрос	Количество страниц (тыс.)
Москва & Лондон	255
Москва & Париж	222
Москва & Париж & Лондон	50

Сколько страниц (в тысячах) будет найдено по запросу *Москва & (Париж | Лондон)*

Решение

Как и в предыдущей задаче, нарисуем диаграмму Эйлера–Венна и посчитаем количество страниц, соответствующее каждой известной области, ограниченной линиями:

Москва & Париж & Лондон(50):



Москва & Лондон (255):



3. Технологии поиска и хранения информации

Москва & Париж (222):



Распределение по областям:



Несложно увидеть, что запросу *Москва & (Париж | Лондон)* соответствует область:



Ответ: 427.

Пример 3.16

В языке запросов поискового сервера для обозначения логической операции «ИЛИ» используется символ «|», а для логической операции «И» — символ «&».

В таблице приведены запросы и количество найденных по ним страниц некоторого сегмента сети Интернет:

Запрос	Найдено страниц (в тысячах)
<i>Леннон & Маккартни & Старр</i>	1100
<i>Леннон & Маккартни & Харрисон</i>	1300
<i>Леннон & Маккартни & Старр & Харрисон</i>	1000

Какое количество страниц (в тыс.) будет найдено по запросу

(Леннон & Маккартни & Старр) / (Леннон & Маккартни & Харрисон)?

Считается, что все запросы выполнялись практически одновременно, так что набор страниц, содержащих все искомые слова, не изменялся за время выполнения запросов.

Решение

Задача кажется на первый взгляд сложной, т.к. содержит 4 ключевых слова, т.е. нам придется рисовать и анализировать диаграмму Эйлера–Венна из 4-х кругов. Это довольно сложно. Но обратим внимание на то, что запрос *Леннон & Маккартни* встречается везде именно в таком виде (нигде не встречается запрос *Леннон* или запрос *Маккартни* отдельно). Поэтому обозначим его одним кругом. Теперь задача свелась к предыдущей и решается аналогично:



Ответ: 1400.

Задача № 7.

Анализ электронных таблиц и диаграмм

Microsoft Excel (в дальнейшем просто — *Excel*) — это программа выполнения расчетов и управления так называемыми электронными таблицами.

Excel позволяет выполнять сложные расчеты, в которых могут использоваться данные, расположенные в разных областях электронной таблицы и связанные между собой определенной зависимостью. Для выполнения таких расчетов в *Excel* существует возможность вводить различные формулы в ячейки таблицы. *Excel* выполняет вычисления и отображает результат в ячейке с формулой.

Важной особенностью использования электронной таблицы является автоматический пересчет результатов при изменении значений ячеек. *Excel* также может строить и обновлять графики, основанные на введенных числах.

Адрес ячейки в электронных таблицах состоит из имени столбца и следующего за ним номера строки, например, C15.

Для написания формул используют адреса ячеек и знаки арифметических операций (+, -, *, /, ^). Формула начинается знаком =.

В *Excel* предусмотрены стандартные функции, которые могут быть использованы в формулах. Это математические,

логические, текстовые, финансовые и другие функции. Однако на экзамене Вам могут встретиться только самые простые функции: СЧЕТ (количество непустых ячеек), СУММ (сумма), СРЗНАЧ (среднее значение), МИН (минимальное значение), МАКС (максимальное значение).

Диапазон ячеек обозначается следующим образом: A1:D4 (все ячейки прямоугольника от A1 до D4).

	A	B	C	D	E
1					
2					
3					
4					
5					
6					

Адреса ячеек бывают относительными, абсолютными и смешанными.

Они по-разному ведут себя при копировании формулы из ячейки в ячейку.

Относительная адресация

Если в ячейке B2 мы напишем формулу $=D1+3$, то таблица воспримет это как «взять значение ячейки на две правее и на одну выше текущей и прибавить к нему 3».

	A	B	C	D	E	F
1						
2		$=D1+3$				
3						
4						
5						

То есть адрес D1 воспринимается таблицей как положение относительно ячейки, куда вводится формула. Такой адрес называется относительным. При копировании такой формулы в другую ячейку таблица автоматически пересчитывает адрес относительно нового расположения формулы:

	A	B	C	D	E	F
1						
2		=D1+3				
3						
4						
5						

	A	B	C	D	E	F
1						
2		3		=F2+3		
3						
4						
5						

Абсолютная адресация

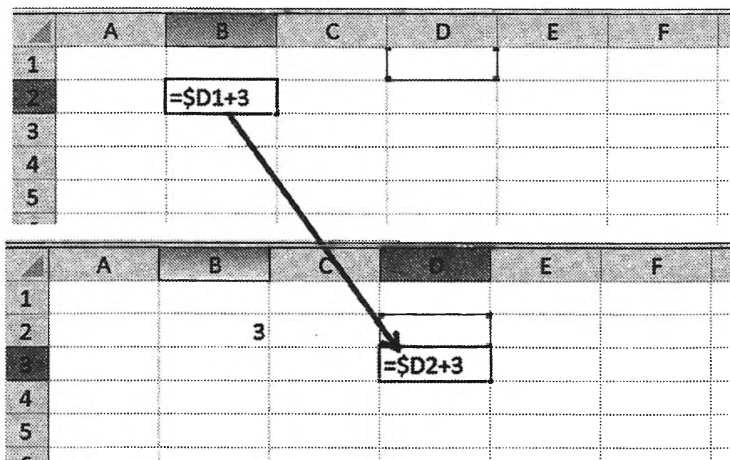
Если нам не нужно, чтобы адрес пересчитывался при копировании формулы, мы можем его «закрепить» в формуле — поставить знак \$ перед буквой и индексом ячейки: $=\$D\$1+3$. Такой адрес называется абсолютным. Такая формула не будет изменяться при копировании:

	A	B	C	D	E	F
1						
2		=\\$D\\$1+3				
3						
4						
5						

	A	B	C	D	E	F
1						
2		3				
3				=\\$D\\$1+3		
4						
5						
6						

Смешанная адресация

Если же мы хотим, чтобы при копировании формулы автоматически пересчитывался, к примеру, только индекс ячейки, а буква оставалась неизменной, мы можем «закрепить» в формуле только букву (или наоборот): $=\$D1+3$. Такой адрес называется смешанным. При копировании формулы будет меняться только индекс в адресе ячейки:



Электронные таблицы. Анализ данных

Пример 4.1

В динамической (электронной) таблице приведены значения пробега автомашин (в км) и общего расхода топлива (в литрах) в четырех транспортных компаниях с 20 по 23 августа.

	20 августа		21 августа		22 августа		23 августа		За четыре дня	
Название компании	Пробег	Расход	Пробег	Расход	Пробег	Расход	Пробег	Расход	Пробег	Расход
Темп	10100	2121	8000	1600	8700	4350	12000	2400	38800	10471
Быстрогруз	500	110	570	228	432	87	700	420	2202	845
Быстрофф	1200	360	2000	800	1500	300	600	240	5300	1700
Ветер	900	450	980	392	300	90	150	30	2330	962



В какой из транспортных компаний средний расход за эти четыре дня был наименьший?

- | | |
|---------------|-------------|
| 1) Темп | 3) Быстрофф |
| 2) Быстрогруз | 4) Ветер |

Решение

Для решения этой задачи нам не нужно никаких специальных знаний. Просто разделим расход за четыре дня на пробег за четыре дня для каждой из фирм:

Темп: $10471/38800=0,27$

Быстрогруз: $845/2202=0,38$

Быстрофф: $1700/5300=0,32$

Ветер: $962/2330=0,41$

Наименьший средний расход (0,27) был у компании Темп.

Ответ: 1.

Пример 4.2

В динамической (электронной) таблице приведены данные о продаже путевок турфирмой «Все на отдых» за 4 месяца. Для каждого месяца вычислено общее количество проданных путевок и средняя цена одной путевки.

Страна	май		июнь		июль		август	
	Продано, шт.	Цена, тыс. руб.	Продано, шт.	Цена, тыс. руб.	Продано, шт.	Цена, тыс. руб.	Продано, шт.	Цена, тыс. руб.
Египет	12	24	15	25	10	22	10	25
Турция	13	27	16	27	12	26	11	28
ОАЭ	12	19	12	22	10	21	9	22
Хорватия	5	30	7	34	13	35	10	33
Продано, шт.	42		50		45		40	
Средняя цена, тыс.руб.		25		27		26		27

Известно, что доход фирмы от продажи каждой путевки не зависит от места отдыха и равен 10% от средней цены путевки в текущем месяце. В каком месяце доход турфирмы был максимальный?

- 1) май
- 2) июнь
- 3) июль
- 4) август

Решение

Поскольку доход фирмы при продаже одной путевки равен 10% от средней стоимости путевки, то доход от продажи всех путевок равен 10% от средней стоимости путевки, умноженным на количество проданных путевок:

Май: $0,1 \cdot 25 \cdot 42 = 105$

Июнь: $0,1 \cdot 27 \cdot 50 = 135$

Июль: $0,1 \cdot 26 \cdot 45 = 117$

Август: $0,1 \cdot 27 \cdot 40 = 108$

Максимальный доход (135) был в июне.

Ответ: 2.

Электронные таблицы. Копирование формул

Пример 4.3

В ячейке C2 записана формула **=E\$3+D2**. Какой вид приобретет формула, после того как ячейку C2 скопируют в ячейку B1?

- 1) **=E\$3+C1**
- 2) **=D\$3+D2**
- 3) **=E\$3+E3**
- 4) **=F\$4+D2**

Решение

Место расположения формулы меняется с C2 на B1, т.е. формула сдвигается на одну ячейку влево и на одну ячейку вверх (буква «уменьшается» на единицу и индекс уменьшается на единицу). Значит, так же изменятся все относитель-

ные адреса, а абсолютные (закрепленные знаком \$) останутся неизменными:

$$= \$E\$3 + C1.$$

Ответ: 1

Пример 4.4

При работе с электронной таблицей в ячейке E3 записана формула **=B2+\$C3**. Какой вид приобретет формула после того, как ячейку E3 скопируют в ячейку D2?

- | | |
|--------------------|--------------------|
| 1) =A1+\$C3 | 2) =A1+\$C2 |
| 3) =E2+\$D2 | 4) =D2+\$E2 |

Решение

Решение аналогично решению предыдущей задачи. Место расположения формулы меняется с E3 на D2, т.е. буква «уменьшается» на 1 и индекс уменьшается на 1. Значит, так же изменятся все относительные адреса, а абсолютные (закрепленные знаком \$) останутся неизменными:

$$=A1+\$C2.$$

Ответ: 2.

Пример 4.5

В ячейке B11 электронной таблицы записана формула. Эту формулу скопировали в ячейку A10. В результате значение в ячейке A10 вычисляется по формуле $x - 3y$, где x — значение в ячейке C22, а y — значение в ячейке D22. Укажите, какая формула могла быть написана в ячейке B11.

- | | |
|--------------------------|--------------------------|
| 1) =C22-3*D22 | 2) =D\$22-3*\$D23 |
| 3) =C\$22-3*D\$22 | 4) =\$C22-3*\$D22 |

Решение

Проанализируем поочередно каждую формулу.

Место расположения формулы меняется с B11 на A10, т.е. буква «уменьшается» на 1 и индекс уменьшается на 1.

Тогда при копировании формулы изменятся следующим образом:

$$=B21-3*C21$$

$$=C\$22-3*\$D22$$

$$=B\$22-3*C\$22$$

$$=\$C21-3*\$D21$$

Условию задачи соответствует формула 2).

Ответ: 2.

Электронные таблицы. Определение значения формулы

Пример 4.6

Дан фрагмент электронной таблицы:

	A	B	C	D
1	1	2	3	
2	4	5	6	
3	7	8	9	

В ячейку D1 введена формула $=\$A\$1*B1+C2$, а затем скопирована в ячейку D2. Какое значение в результате появится в ячейке D2?

- 1) 10 2) 14 3) 16 4) 24

Решение

Место расположения формулы меняется с D1 на D2, т.е. буква не меняется, а индекс увеличивается на 1.

Значит, формула примет вид: $=\$A\$1*B2+C3$. Подставим в формулу числовые значения ячеек: $1*5+9=14$. Правильный ответ указан под номером 2.

Ответ: 2.

Пример 4.7

В электронной таблице значение формулы **=СРЗНАЧ(A6:C6)** равно (-2). Чему равно значение формулы **=СУММ(A6:D6)**, если значение ячейки D6 равно 5?

- 1) 1 2) -1 3) -3 4) 7

Решение

	A	B	C	D
1				
2				
3				
4				
5				
6				5

По определению среднего значения:

$$\text{СРЗНАЧ}(A6:C6) = \text{СУММ}(A6:C6)/3 = -2$$

Значит, $\text{СУММ}(A6:C6) = -6$

$$\text{СУММ}(A6:D6) = \text{СУММ}(A6:C6) + D6 = -6 + 5 = -1$$

Ответ: 2.

Пример 4.8

В электронной таблице значение формулы **=СРЗНАЧ(A1:B4)** равно 3. Чему равно значение ячейки A4, если значение формулы **=СУММ(A1:B3)** равно 30, а значение ячейки B4 равно 5?

- 1) -11 2) 11 3) 4 4) -9

Решение:

	A	B
1		
2		
3		
4		5

$$\text{СРЗНАЧ}(A1:B4) = \text{СУММ}(A1:B4)/8 = 3$$

$$\text{СУММ}(A1:B4) = 24 = \text{СУММ}(A1:B3) + A4 + B4 = 30 + A4 + 5$$

$$A4 = 24 - 35 = -11.$$

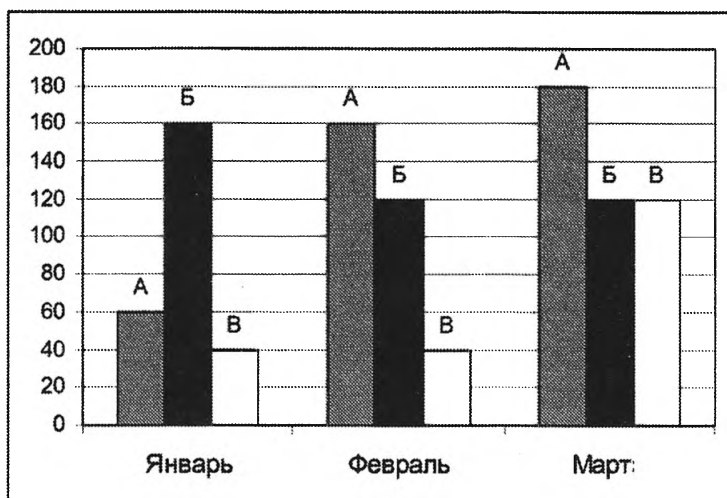
Ответ: 1.

Диаграммы.

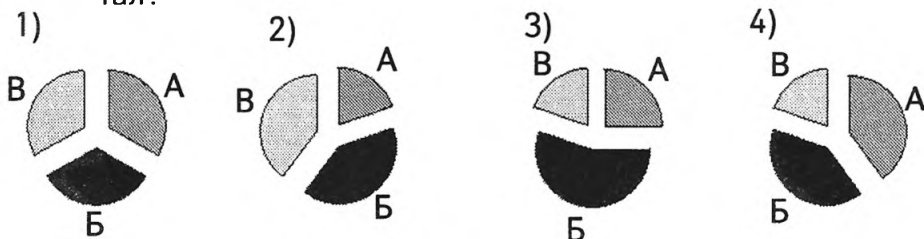
Сравнение разных видов диаграмм

Пример 4.9

На диаграмме показаны объемы выпуска продукции трех видов (А, Б и В) за каждый месяц первого квартала:



Какая из диаграмм правильно отражает соотношение объемов выпуска этих видов продукции за весь квартал?



Решение

Просуммируем объемы выпуска за 3 месяца для каждого вида продукции на основании первой диаграммы:

А $60 + 160 + 180 = 400$;

Б $160 + 120 + 120 = 400$;

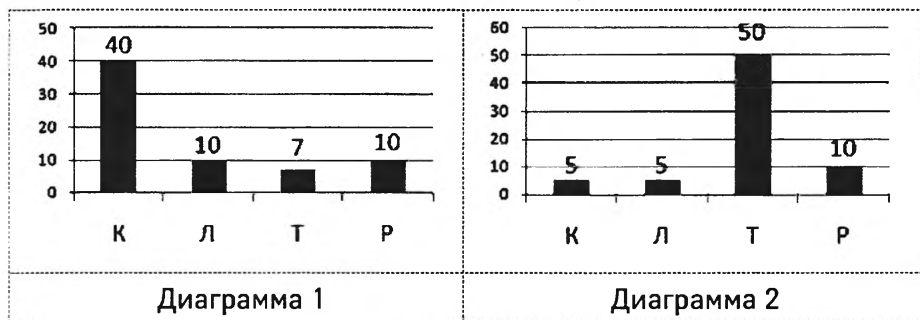
В $40 + 40 + 120 = 200$.

То есть правильная диаграмма должна иметь два одинаковых сектора (400) А и Б, а сектор В вдвое меньший (200). Такая диаграмма расположена под номером 4.

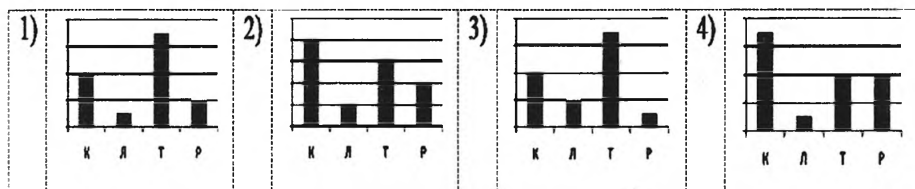
Ответ: 4.

Пример 4.10

На диаграмме 1 показано количество (в штуках) проданных канцтоваров за 1 день, а на диаграмме 2 — цена (в рублях за штуку) каждого из проданных товаров. Здесь К — карандаши, Л — ластики, Т — точилки и Р — ручки:



Какая диаграмма правильно показывает распределение выручки от товаров разного типа:



Решение

Выручка равна цене, умноженной на количество. На основании первых двух диаграмм посчитаем выручку для каждого типа товаров:

$$К \quad 40 \cdot 5 = 200;$$

$$Л \quad 10 \cdot 5 = 50;$$

$$Т \quad 7 \cdot 50 = 350;$$

$$Р \quad 10 \cdot 10 = 100.$$

Очевидно, что эти данные отражены на диаграмме 1.

Ответ: 1.

Диаграммы. Анализ данных

Пример 4.11

В соревнованиях по зимним видам спорта принимают участие лыжники (Л), конькобежцы (К) и хоккеисты (Х). Спортсмены имеют разный уровень мастерства: каждый имеет либо III, либо II, либо I разряд, либо является мастером спорта (М). На диаграмме 1 отражено количество спортсменов с различным уровнем спортивного мастерства, а на диаграмме 2 — распределение спортсменов по видам спорта.

Диаграмма 1

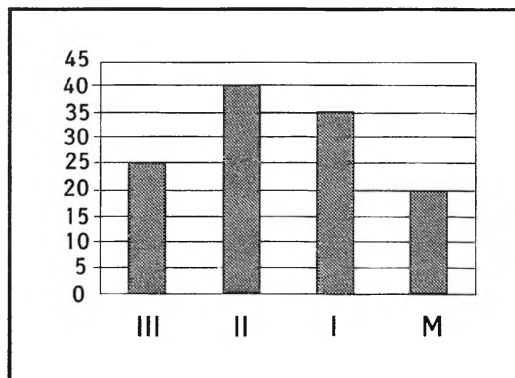
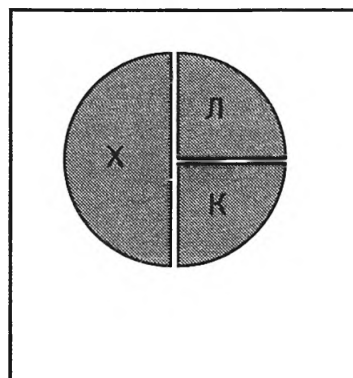


Диаграмма 2



Имеются 4 утверждения:

- А) Все спортсмены, имеющие I разряд, могут являться конькобежцами.
- Б) Все лыжники могут быть мастерами спорта.
- В) Все хоккеисты могут иметь II разряд.
- Г) Все спортсмены, имеющие I разряд, могут являться хоккеистами.

Какое из этих утверждений следует из анализа обеих представленных диаграмм?

- 1) А 2) Б 3) В 4) Г

Решение

На основании диаграммы 1 посчитаем общее количество спортсменов: $25 + 40 + 35 + 20 = 120$.

Проверим поочередно каждое из утверждений.

Доля спортсменов, имеющих I разряд, $35/120 \approx 0,29$. Это больше, чем доля конькобежцев (0,25). Значит, все I-разрядники не могут быть конькобежцами.

Б) Доля мастеров спорта $20/120 = 1/6$, а доля лыжников $1/4$, значит, все лыжники не могут быть мастерами спорта.

В) Доля хоккеистов $1/2 = 0,5$. Доля спортсменов, имеющих II разряд $40/120 = 1/3$. Значит, все хоккеисты не могут иметь II разряд.

Г) Доля спортсменов, имеющих I разряд, $35/120 \approx 0,29$. Доля хоккеистов 0,5. Значит, все спортсмены, имеющие I разряд, могут являться хоккеистами.

Ответ: 4.

Электронные таблицы и диаграммы

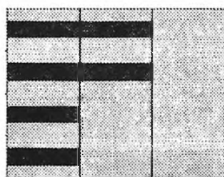
Пример 4.12

Дан фрагмент электронной таблицы в режиме отображения формул.

	A	B	C	D
1	$=C2-B1$	$=B2-C2$	$=B1+C2$	$=(C1-C2)*3$
2		3	2	

После выполнения вычислений построили диаграмму по значениям диапазона A1:D1. Укажите полученную диаграмму:

1)



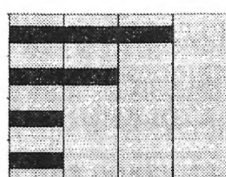
2)



3)



4)



Решение

Вычислим по формулам значения ячеек A1:D1.

$$B1 = 3 - 2 = 1;$$

$$A1 = 2 - 1 = 1;$$

$$C1 = 1 + 2 = 3;$$

$$D1 = 1 \cdot 3 = 3.$$

Этим данным соответствует диаграмма 3.

Ответ: 3.

Задача № 12

Организация компьютерных сетей. Адресация в Интернете

Адрес документа в Интернете состоит из следующих частей:

Протокол (чаще всего **http** или **ftp**), последовательность символов «://», доменное имя сайта, каталог на сервере, где находится файл, имя файла. Каталоги разделяются символом «/».

Например: <http://www.hs.ru/files/user/olga/filenew.zip>

IP-адрес компьютера имеет длину 4 байта. Для удобства IP-адрес записывают в виде четырех чисел, разделенных точками. Числа принимают значения от 0 до 255 (т.к. 255 — 8 единиц в двоичной системе — наибольшее число, которое можно записать в один байт).

IP-адрес состоит из двух частей: адреса сети и номера компьютера в этой сети. Для деления адреса на части используют маску. Маска — это 32-битное число, в двоичной записи которого сначала стоят единицы, а потом — нули. Единицы определяют часть адреса, относящуюся к адресу сети, а нули — часть адреса, относящуюся к номеру компьютера в сети.

Адрес файла в Интернете

Пример 5.1

Доступ к файлу **ftp.net**, находящемуся на сервере **txt.org**, осуществляется по протоколу **http**. В таблице фрагменты адреса файла закодированы буквами от А до Ж. Запишите последовательность этих букв, кодирующую адрес указанного файла в сети Интернет.

А	.net
Б	ftp
В	://
Г	http
Д	/
Е	.org
Ж	txt

Решение

При записи адреса файла в Интернете сначала указывается протокол, затем ставится последовательность символов **://**, затем имя сервера, затем символ **/** и лишь потом имя файла: **http://txt.org/ftp.net**.

Ответ: ГВЖЕДБА.

Пример 5.2

Идентификатор некоторого ресурса сети Интернет имеет следующий вид: **http://www.ftp.ru/index.html**

Какая часть этого идентификатора указывает на протокол, используемый для передачи ресурса?

1) **www** 2) **ftp** 3) **http** 4) **html**

Решение

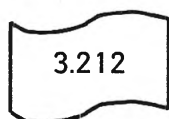
Протокол — самая первая часть адреса, стоящая до двоеточия: **http**.

Ответ: 3.

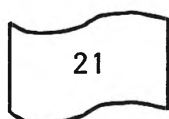
Восстановление IP-адресов

Пример 5.3

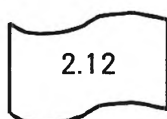
Петя записал IP-адрес школьного сервера на листке бумаги и положил его в карман куртки. Петина мама случайно постирала куртку вместе с запиской. После стирки Петя обнаружил в кармане четыре обрывка с фрагментами IP-адреса. Эти фрагменты обозначены буквами А, Б, В и Г. Восстановите IP-адрес. В ответе укажите последовательность букв, обозначающих фрагменты, в порядке, соответствующем IP-адресу.



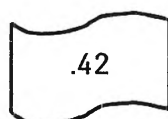
А



Б



В



Г

Решение

IP-адрес представляет собой 4 числа, разделенные точками, причем эти числа не больше 255.

Посмотрим внимательнее на данные фрагменты: под буквой Г мы видим «.42». Так как числа в IP-адресе не могут быть больше 255, мы не можем ничего дописать к этому числу, а фрагментов, начинающихся с точки, больше нет, следовательно, этот фрагмент — последний.

На фрагменте под буквой Б число без точек, значит, это либо последний фрагмент, либо первый. Место последнего фрагмента уже занято, значит, фрагмент Б первый.

В конце фрагмента А — число 212, отделенное точкой, значит, за фрагментом А должен следовать фрагмент, начинающийся с точки. Значит, фрагмент А идет перед фрагментом Г.

Ответ: БВАГ.

Определение адреса сети

Пример 5.4.

В терминологии сетей TCP/IP маской сети называется двоичное число, определяющее, какая часть IP-адреса узла сети относится к адресу сети, а какая — к адресу самого узла в этой сети. Обычно маска записывается по тем же правилам, что и IP-адрес. Адрес сети получается в результате применения поразрядной конъюнкции к заданным IP-адресу узла и маске.

По заданным IP-адресу узла и маске определите адрес сети.

IP-адрес узла: 218.137.218.137

Маска: 255.255.248.0

При записи ответа выберите из приведённых в таблице чисел четыре элемента IP-адреса и запишите в нужном порядке соответствующие им буквы без использования точек.

При записи ответа выберите из приведенных в таблице чисел 4 фрагмента четыре элемента IP-адреса и запишите в нужном порядке соответствующие им буквы без точек.

A	B	C	D	E	F	G	H
255	249	218	216	137	32	8	0

Пример. Пусть искомый адрес сети 192.168.128.0 и дана таблица

A	B	C	D	E	F	G	H
128	168	255	8	127	0	17	192

В этом случае правильный ответ будет HBAF.

Решение

Адрес сети получается в результате поразрядной конъюнкции чисел маски и чисел адреса узла (в двоичном коде). Конъюнкция 0 с любым числом всегда равна 0, а конъюнкция 255_{10} (8 единиц в двоичной системе) с любым числом равна этому числу.

IP-адрес узла: 218.137.218.137

Маска: 255.255.248.0

Значит, первые два числа адреса сети останутся такими же, как у IP-адреса узла, а последнее число будет 0. Нам осталось провести поразрядную конъюнкцию двоичной записи чисел 218 и 248.

$$248_{10} = 11111000_2.$$

$$218_{10} = 11011010_2.$$

Результатом конъюнкции является число $11011000_2 = 216$.

Сопоставим варианты ответа получившимся числам: 218, 137, 216, 0.

Ответ: CEDH.

Определение маски сети

Пример 5.5

В терминологии сетей TCP/IP маской сети называется двоичное число, определяющее, какая часть IP-адреса узла сети относится к адресу сети, а какая — к адресу самого узла в этой сети. Обычно маска записывается по тем же правилам, что и IP-адрес, — в виде четырёх байтов, причём каждый байт записывается в виде десятичного числа. При этом в маске сначала (в старших разрядах) стоят единицы, а затем с некоторого разряда — нули.

Адрес сети получается в результате применения поразрядной конъюнкции к заданному IP-адресу узла и маске.

Например, если IP-адрес узла равен 231.32.255.131, а маска равна 255.255.240.0, то адрес сети равен 231.32.240.0.

Для узла с IP-адресом 111.81.208.27 адрес сети равен 111.81.192.0. Чему равно наименьшее возможное значение третьего слева байта маски? Ответ запишите в виде десятичного числа.

Решение

Поскольку нас интересует только третий байт маски, запишем третий байт IP-адреса и адреса сети в двоичной системе счисления:

$$208_{10} = 11010000_2.$$

$$192_{10} = 11000000_2.$$

С каким числом нужно произвести конъюнкцию 11010000_2 , чтобы получить 11000000_2 ? Очевидно, что первые две цифры должны быть единицами, а 4-я нулем.

Это или 11000000 , или 11100000 . По условию задачи требуется найти наименьшее значение — это 11000000 .

$$11000000_2 = 192_{10}.$$

Ответ: 192.

Подсчет количества адресов

Пример 5.6

В терминологии сетей TCP/IP маской подсети называется 32-разрядное двоичное число, определяющее, какие именно разряды IP-адреса компьютера являются общими для всей подсети — в этих разрядах маски стоит 1. Обычно маски записываются в виде четверки десятичных чисел — по тем же правилам, что и IP-адреса. Для некоторой подсети используется маска 255.255.254.0. Сколько различных адресов компьютеров теоретически допускает эта маска, если два адреса (адрес сети и широковещательный) не используют?

Решение

За адрес компьютера в маске отвечают разряды, содержащие нули. В маске 255.255.254.0. первые два числа состоят полностью из единиц, т.е. определяют адрес сети. Запишем третье число маски в двоичном виде: $254 = 11111110_2$.

Четвертое число маски в двоичном представлении состоит из 8 нулей.

То есть маска выглядит следующим образом:

11111111 11111111 11111110 00000000.

То есть под адрес компьютера выделено 9 разрядов, значит, туда можно записать $2^9 = 512$ адресов, но, так как два адреса не используются, получаем $512 - 2 = 510$.

Ответ: 510.

Определение номера компьютера в сети

Пример 5.7

Маской подсети называется 32-разрядное двоичное число, которое определяет, какая часть IP-адреса компьютера относится к адресу сети, а какая часть IP-адреса определяет адрес компьютера в подсети. В маске подсети старшие биты, отведенные в IP-адресе компьютера для адреса сети, имеют значение 1; младшие биты, отведенные в IP-адресе компьютера для адреса компьютера в подсети, имеют значение 0.

Если маска подсети 255.255.224.0 и IP-адрес компьютера в сети 206.158.124.67, то номер компьютера в сети равен _____

Решение

Первые два числа маски равны 255 (в двоичной записи состоят полностью из единиц). Третье число маски $224_{10} = 11100000_2$. Четвертое число маски состоит из 8 нулей. То есть маска выглядит следующим образом:

● Информатика

11111111 11111111 11100000 00000000

То есть под адрес компьютера отведено 13 разрядов.

Запишем последние два числа IP-адреса компьютера в сети: $124_{10} = 1111100_2$.

$$67_{10} = 1000011_2$$

То есть последние два числа IP-адреса компьютера в сети записываются так:

01111100 01000011.

Нам нужны только последние 13 разрядов (подчеркнутая часть), переведем ее в десятичную систему счисления: $1110001000011_2 = 7235_{10}$.

Ответ: 7235.

Задача № 5

Кодирование и декодирование информации

Кодирование — это перевод информации, представленной символами первичного алфавита, в последовательность кодов.

Декодирование (операция, обратная кодированию) — перевод кодов в набор символов первичного алфавита.

Кодирование может быть *равномерное* и *неравномерное*. При равномерном кодировании каждый символ исходного алфавита заменяется кодом одинаковой длины. При неравномерном кодировании разные символы исходного алфавита могут заменяться кодами разной длины.

Код называется однозначно декодируемым, если любое сообщение, составленное из кодовых слов, можно декодировать единственным способом.

Равномерное кодирование всегда однозначно декодируемо.

Для неравномерных кодов существует следующее достаточное (но не необходимое) условие однозначного декодирования:

Сообщение однозначно декодируемо с начала, если выполняется условие **Фано**: **никакое кодовое слово не является началом другого кодового слова**.

Сообщение однозначно декодируемо с конца, если выполняется обратное условие **Фано**: **никакое кодовое слово не является окончанием другого кодового слова**.

Кодирование в различных системах счисления

Пример 6.1

Для кодирования букв О, В, Д, П, А решили использовать двоичное представление чисел 0, 1, 2, 3 и 4 соответственно (с сохранением одного незначащего нуля в случае одноразрядного представления). Если закодировать последовательность букв ВОДОПАД таким способом и результат записать восьмеричным кодом, то получится

- 1) 22162
- 2) 1020342
- 3) 2131453
- 4) 34017

Решение

Представим коды указанных букв в двоичном коде, добавив незначащий нуль для одноразрядных чисел:

О	В	Д	П	А
0	1	2	3	4
00	01	10	11	100

Закодируем последовательность букв:

ВОДОПАД — 010010001110010.

Разобьем это представление на тройки справа налево и переведем каждую тройку в восьмеричное число.

$$010\ 010\ 001\ 110\ 010_2 = 22162_8.$$

Правильный ответ указан под номером 1.

Ответ: 1.

Пример 6.2

Для передачи по каналу связи сообщения, состоящего только из символов А, Б, В и Г, используется посимвольное кодирование: А-10, Б-11, В-110, Г-0. Через канал связи передаётся сообщение: ВАГБААГВ. Закодируйте сообщение данным кодом. Полученное двоичное число переведите в шестнадцатеричный вид.

- 1) D3A6
- 2) 62032206
- 3) 6A3D
- 4) CADBAADC

Решение

Закодируем последовательность букв: ВАГБААГВ — 1101001110100110. Разобьем это представление на четвёрки справа налево и переведем каждую четверку в шестнадцатеричное число:

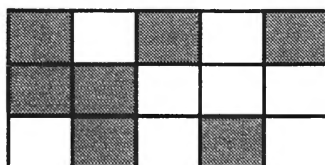
$$1101\ 0011\ 1010\ 0110_2 = D3A6_{16}.$$

Правильный ответ указан под номером 1.

Ответ: 1.

Пример 6.3

Черно-белое растровое изображение кодируется построчно, начиная с левого верхнего угла и заканчивая в правом нижнем углу. При кодировании 1 обозначает черный цвет, а 0 — белый.



Для компактности результат записали в восьмеричной системе счисления. Выберите правильную запись кода.

- 1) 57414
- 2) 53414
- 3) 53412
- 4) 53012

Решение

Код первой строки: 10101.

Код второй строки: 11000.

Код третьей строки: 01010.

Запишем коды по порядку в одну строку: 101011100001010.

Теперь разобьем это представление на тройки справа налево и переведем каждую тройку в восьмеричное число:

$$101\ 011\ 100\ 001\ 010_2 = 53412_8.$$

Правильный ответ указан под номером 3.

Ответ: 3.

Расшифровка сообщений

Пример 6.4

Для 5 букв латинского алфавита заданы их двоичные коды (для некоторых букв — из двух бит, для некоторых — из трех). Эти коды представлены в таблице:

a	b	c	d	e
100	110	011	01	10

Определите, какой набор букв закодирован двоичной строкой 1000110110110, если известно, что все буквы в последовательности — разные:

- 1) cbade
- 2) acdeb
- 3) acbed
- 4) bacde

Решение

Мы видим, что условие Фано и обратное условие Фано не выполняются, значит, код можно раскодировать неоднозначно.

Значит, будем перебирать варианты, пока не получим подходящее слово :

1) 100 011 01 10 110.

Первая буква определяется однозначно, ее код 100: а.

Пусть вторая буква — с, тогда следующая буква — d, потом — е и b.

Такой вариант удовлетворяет условию, значит, окончательно получили ответ: acdeb.

Ответ: 2.

Пример 6.5

Для передачи чисел по каналу с помехами используется код проверки четности. Каждая его цифра записывается в двоичном представлении, с добавлением ведущих нулей до длины 4, и к получившейся последовательности дописывается сумма ее элементов по модулю 2 (например, если передаем 23, то получим последовательность 0010100110). Определите, какое число передавалось по каналу в виде 01010100100111100011?

1) 59143

2) 5971

3) 102153

4) 10273

Решение

При кодировании под саму цифру отводится 4 разряда, и еще один под код проверки четности (т.е. всего под цифру отводится 5 разрядов).

Разобьем заданную последовательность на группы по 5 бит в каждой:

01010, 10010, 01111, 00011;

отбросим пятый (последний) бит в каждой группе:

0101, 1001, 0111, 0001.

Это и есть двоичные коды передаваемых чисел:

$0101_2 = 5$, $1001_2 = 9$, $0111_2 = 7$, $0001_2 = 1$.

Таким образом, были переданы числа 5, 9, 7, 1 или число 5971.

Ответ: 2.

Пример 6.6.

По каналу связи передаются сообщения, содержащие только 4 буквы — П, О, Р, Т. Для кодирования букв используются 5-битовые кодовые слова:

П — 11111, О — 11000, Р — 00100, Т — 00011.

Для этого набора кодовых слов выполнено такое свойство: любые два слова из набора отличаются не менее чем в трех позициях.

Это свойство важно для расшифровки сообщений при наличии помех (в предположении, что передаваемые биты могут искажаться, но не пропадают). Закодированное сообщение считается принятым корректно, если его длина кратна 5 и каждая пятёрка отличается от некоторого кодового слова не более чем в одной позиции; при этом считается, что пятёрка кодирует соответствующую букву. Например, если принята пятерка 00000, то считается, что передавалась буква Р.

Среди приведенных ниже сообщений найдите то, которое принято корректно, и укажите его расшифровку (пробелы несущественны).

11011 11100 00011 11000 01110

00111 11100 11110 11000 00000

1) ПОТОП

2) РОТОР

3) ТОПОР

4) ни одно из сообщений не принято корректно

Решение

Длина обоих сообщений кратна пяти.

Анализируя первое сообщение «11011 11100 00011 11000 01110», приходим к выводу, что оно принято некорректно, поскольку нет такого слова, которое бы отличалось от слова «01110» только в одной позиции.

Рассмотрим второе сообщение. Учитывая, что каждая пятёрка отличается от некоторого кодового слова не более чем в одной позиции, его возможно расшифровать только как «ТОПОР».

Ответ: 3.

Пример 6.7

Для передачи данных по каналу связи используется 5-битовый код. Сообщение содержит только буквы А, Б и В, которые кодируются следующими кодовыми словами: А — 11010, Б — 10111, В — 01101.

При передаче возможны помехи. Однако некоторые ошибки можно попытаться исправить. Любые два из этих трёх кодовых слов отличаются друг от друга не менее чем в трёх позициях. Поэтому если при передаче слова произошла ошибка не более чем в одной позиции, то можно сделать обоснованное предположение о том, какая буква передавалась. (Говорят, что «код исправляет одну ошибку».) Например, если получено кодовое слово 10110, считается, что передавалась буква Б. (Отличие от кодового слова для Б только в одной позиции, для остальных кодовых слов отличий больше.) Если принятое кодовое слово отличается от кодовых слов для букв А, Б, В более чем в одной позиции, то считается, что произошла ошибка (она обозначается 'х'). Получено сообщение 11000 11101 10001 11111. Декодировать это сообщение — выберите правильный вариант.

- | | |
|---------|---------|
| 1) АххБ | 3) хххх |
| 2) АВхБ | 4) АВББ |

Решение

Декодируем каждое слово сообщения. Первое слово: 11000 отличается от буквы А только одной позицией. Второе слово: 11101 отличается от буквы В только одной позицией. Третье слово: 10001 отличается от любой буквы более чем одной позицией. Четвёртое слово: 11111 отличается от буквы Б только одной позицией.

Таким образом, ответ: АВхБ.

Ответ: 2.

Однозначное кодирование**Пример 6.8**

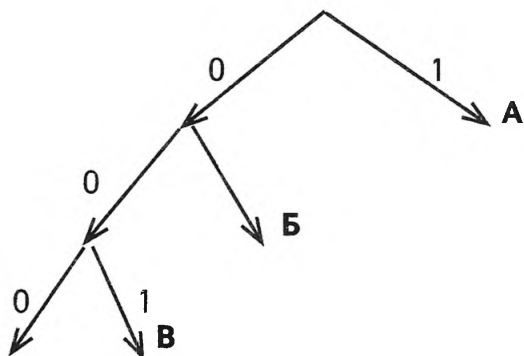
Для передачи по каналу связи сообщения, состоящего только из букв А, Б, В, Г, решили использовать неравномерный по длине код: А=1, Б=01, В=001. Как нужно закодировать букву Г, чтобы длина кода была минимальной и допускалось однозначное разбиение кодированного сообщения на буквы?

- 1) 0001
- 2) 000
- 3) 11
- 4) 101

Решение

Для анализа соблюдения условия однозначного декодирования (условия Фано) изобразим коды в виде дерева. Тогда однозначность выполняется, если каждая буква является листом дерева.

Видим, что ближайший от корня дерева свободный лист (т.е. код с минимальной длиной) имеет код 000.



Ответ: 2.

Пример 6.9

Для кодирования некоторой последовательности, состоящей из букв У, Ч, Е, Н, И и К, используется неравномерный двоичный префиксный код. Вот этот код: У — 000, Ч — 001, Е — 010, Н — 100, И — 011, К — 11. Можно ли сократить для одной из букв длину кодового слова так, чтобы код по-прежнему остался префиксным? Коды остальных букв меняться не должны.

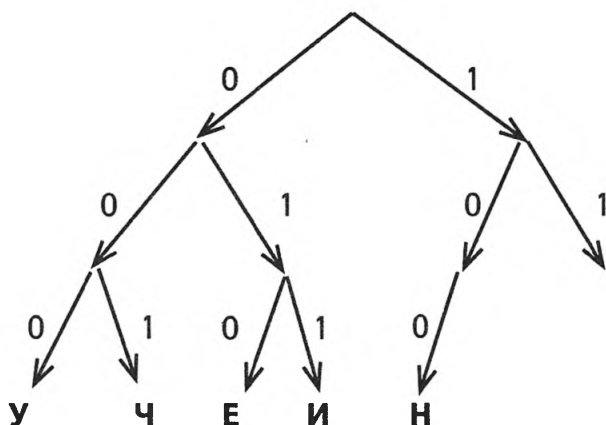
Выберите правильный вариант ответа.

Примечание. Префиксный код — это код, в котором ни одно кодовое слово не является началом другого; такие коды позволяют однозначно декодировать полученную двоичную последовательность.

- 1) кодовое слово для буквы Е можно сократить до 01
- 2) кодовое слово для буквы К можно сократить до 1
- 3) кодовое слово для буквы Н можно сократить до 10
- 4) это невозможно

Решение

Для анализа соблюдения условия однозначного декодирования (условия Фано) изобразим коды в виде дерева. Тогда однозначность выполняется, если каждая буква является листом дерева:



Легко заметить, что если букву Н перенести в вершину 10, она останется листом. То есть кодовое слово для буквы Н можно сократить до 10.

Правильный ответ указан под номером 3.

Ответ: 3.

Пример 6.10

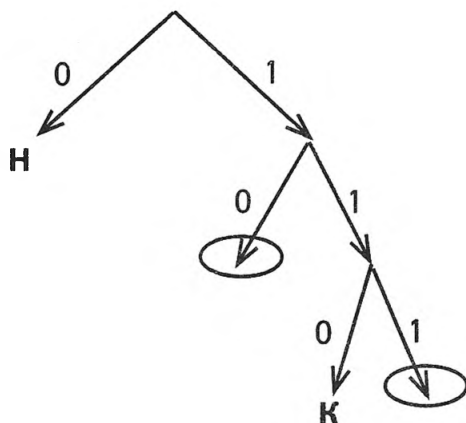
Для кодирования некоторой последовательности, состоящей из букв К, Л, М, Н, решили использовать неравномерный двоичный код, удовлетворяющий условию Фано. Для буквы Н использовали кодовое слово 0, для буквы К — кодовое слово 110. Какова наименьшая возможная суммарная длина всех четырёх кодовых слов?

- 1) 7
- 2) 8
- 3) 9
- 4) 10

Примечание. Условие Фано означает, что никакое кодовое слово не является началом другого кодового слова. Это обеспечивает возможность однозначной расшифровки закодированных сообщений.

Решение

Для анализа соблюдения условия однозначного декодирования (условия Фано) изобразим коды в виде дерева. Тогда однозначность выполняется, если каждая буква является листом дерева:



Легко заметить, что ближайшие от корня свободные листы (т.е. свободные коды с наименьшей длиной) — это 10 и 111. Таким образом, наименьшая возможная суммарная длина всех четырёх кодовых слов будет $1 + 3 + 2 + 3 = 9$.

Правильный ответ указан под номером 3.

Ответ: 3.

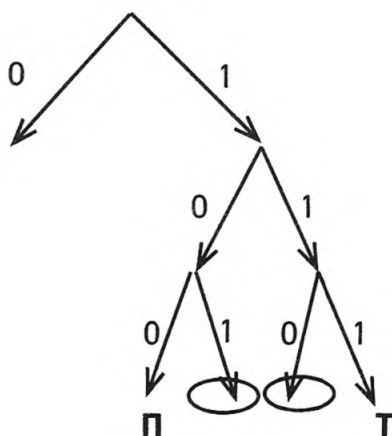
Пример 6.11

По каналу связи передаются сообщения, содержащие только четыре буквы: П, О, С, Т; для передачи используется двоичный код, допускающий однозначное декодирование. Для букв Т, О, П используются такие кодовые слова: Т: 111, О: 0, П: 100.

Укажите кратчайшее кодовое слово для буквы С, при котором код будет допускать однозначное декодирование. Если таких кодов несколько, укажите код с наименьшим числовым значением.

Решение

Для анализа соблюдения условия однозначного декодирования (условия Фано) изобразим коды в виде дерева. Тогда однозначность выполняется, если каждая буква является листом дерева:



Легко заметить, что ближайшие от корня свободные листы (т.е. свободные коды с наименьшей длиной) — это 101 и 110. Наименьшее числовое значение имеет код 101.

Ответ: 101.

Задача № 9

Кодирование и передача звуковой и графической информации

При оцифровке звука в памяти запоминаются только отдельные значения сигнала. Чем чаще записывается сигнал, тем лучше качество записи.

Частота дискретизации f — это количество преобразований аналогового сигнала в цифровой за одну секунду. Измеряется в Герцах (Гц).

Глубина кодирования (а также разрешение) — это количество бит, выделяемое на одно преобразование сигнала. Измеряется в битах (бит).

Возможна запись нескольких каналов: одного (моно), двух (стерео), четырех (квадро).

Обозначим частоту дискретизации — f (Гц), глубину кодирования — B (бит), количество каналов — k , время записи — t (с).

Количество уровней дискретизации d можно рассчитать по формуле: $d = 2^B$.

Тогда объем записанного файла

$$V(\text{бит}) = f \cdot B \cdot k \cdot t.$$

Или, если нам дано количество уровней дискретизации,

$$V(\text{бит}) = f \cdot \log_2 d \cdot k \cdot t.$$

Единицы измерения объемов информации:

$$1 \text{ б (байт)} = 8 \text{ бит}$$

$$1 \text{ Кб (килобайт)} = 2^{10} \text{ б}$$

$$1 \text{ Мб (мегабайт)} = 2^{20} \text{ б}$$

$$1 \text{ Гб (гигабайт)} = 2^{30} \text{ б}$$

$$1 \text{ Тб (терабайт)} = 2^{40} \text{ б}$$

$$1 \text{ Пб (петабайт)} = 2^{50} \text{ б}$$

При оцифровке графического изображения качество картинки зависит от количества точек и количества цветов, в которые можно раскрасить точку.

Если X — количество точек по горизонтали,

Y — количество точек по вертикали,

I — глубина цвета (количество бит, отводимых для кодирования одной точки),

то количество различных цветов в палитре $N = 2^I$. Соответственно, $I = \log_2 N$.

Тогда объем файла, содержащего изображение,

$$V(\text{бит}) = X \cdot Y \cdot I$$

Или, если нам дано количество цветов в палитре,

$$V(\text{бит}) = X \cdot Y \cdot \log_2 N.$$

Скорость передачи информации по каналу связи (пропускная способность канала) вычисляется как количество информации в битах, переданное за 1 секунду (бит/с).

Объем переданной информации вычисляется по формуле

$$V = q \cdot t,$$

где q — пропускная способность канала, а t — время передачи.

Кодирование звука

Пример 6.12

Производится двухканальная (стерео) звукозапись с частотой дискретизации 16 кГц и глубиной кодирования 32 бит. Запись длится 12 минут, ее результаты записываются в файл, сжатие данных не производится. Какое из приведенных ниже чисел наиболее близко к размеру полученного файла, выраженному в мегабайтах?

1) 30

2) 45

3) 75

4) 90

Решение

$$V(\text{бит}) = f(\text{Гц}) \cdot B(\text{бит}) \cdot k \cdot t(\text{с}),$$

где V — размер файла, f — частота дискретизации, B — глубина кодирования, k — количество каналов, t — время.

Значит, $V(\text{Мб}) = (f \cdot B \cdot k \cdot t) / 2^{23}$.

Переведем все величины в требуемые единицы измерения:

$$I(\text{Мб}) = (16 \cdot 1000 \cdot 32 \cdot 2 \cdot 12 \cdot 60) / 2^{23}.$$

Представим все возможные числа как степени двойки:

$$\begin{aligned} I(\text{Мб}) &= (2^4 \cdot 2^3 \cdot 125 \cdot 2^5 \cdot 2 \cdot 2^2 \cdot 3 \cdot 15 \cdot 2^2) / 2^{23} = \\ &= (5625 \cdot 2^{17}) / 2^{23} = 5625 / 2^6 = 5625 / 64 \approx 90. \end{aligned}$$

Ответ: 4.

Без представления чисел через степени двойки вычисления становятся намного сложнее.

Частота — это физическая величина, а потому 16 кГц = 16 · 1000 Гц, а не 16 · 2¹⁰. Иногда этой разницей можно пренебречь, но на последних диагностических работах она влияла на правильность ответа.

Пример 6.13

Производится одноканальная (моно) звукозапись с частотой дискретизации 128 Гц. При записи использовались 64 уровня дискретизации. Запись длится 6 минут 24 секунды, её результаты записываются в файл, причём каждый сигнал кодируется минимально возможным и одинаковым количеством битов. Какое из приведённых ниже чисел наиболее близко к размеру полученного файла, выраженному в килобайтах?

- 1) 24 2) 36 3) 128 4) 384

Решение

$$V(\text{бит}) = f \cdot \log_2 d \cdot k \cdot t,$$

где V — размер файла, f — частота дискретизации, d — количество уровней дискретизации, k — количество каналов, t — время.

Переведем все величины в требуемые единицы измерения и представим все возможные числа как степени двойки:

$$V(\text{Кб}) = (f \cdot \log_2 d \cdot k \cdot t) / 2^{13} = (128 \cdot \log_2 64 \cdot 1 \cdot 384) / 2^{13} = \\ = (2^7 \cdot 6 \cdot 3 \cdot 2^7) / 2^{13} = 9 \cdot 4 = 36.$$

Ответ: 2.

Пример 6.14

В течение трех минут производилась четырёхканальная (квадро) звукозапись с частотой дискретизации 16 кГц и 24-битным разрешением. Сжатие данных не производилось. Какая из приведенных ниже величин наиболее близка к размеру полученного файла?

- 1) 25 Мбайт
- 2) 35 Мбайт
- 3) 45 Мбайт
- 4) 55 Мбайт

Решение

$$V(\text{бит}) = f(\text{Гц}) \cdot B(\text{бит}) \cdot k \cdot t(\text{с}),$$

где V — размер файла, f — частота дискретизации, B — глубина кодирования (или разрешение), k — количество каналов, t — время.

Значит,

$$V(\text{Мб}) = (f \cdot B \cdot k \cdot t) / 2^{23} = \\ = (16 \cdot 1000 \cdot 24 \cdot 4 \cdot 3 \cdot 60) / 2^{23} = \\ = (2^4 \cdot 2^3 \cdot 125 \cdot 3 \cdot 2^3 \cdot 2^2 \cdot 3 \cdot 15 \cdot 2^2) / 2^{23} = \\ = (125 \cdot 9 \cdot 15 \cdot 2^{14}) / 2^{23} = 16875 / 2^9 = 32,96 \approx 35.$$

Ответ: 2.

Пример 6.15

Музыкальный фрагмент был записан в формате моно, оцифрован и сохранен в виде файла без использования сжатия данных. Размер полученного файла —

24 Мбайт. Затем тот же музыкальный фрагмент был записан повторно в формате стерео (двухканальная запись) и оцифрован с разрешением в 4 раза выше и частотой дискретизации в 1,5 раза меньше, чем в первый раз. Сжатие данных не производилось. Укажите размер файла в Мбайт, полученного при повторной записи. В ответе запишите только целое число, единицу измерения писать не нужно.

Решение

$$V(\text{бит}) = f(\text{Гц}) \cdot B(\text{бит}) \cdot k \cdot t(\text{с}),$$

где V — размер файла, f — частота дискретизации, B — глубина кодирования, k — количество каналов, t — время.

Тогда результатом первой записи стал файл размером $24 \text{ Мб} = (f \cdot B \cdot 1 \cdot t) / 2^{23}$.

При повторной записи частота дискретизации стала $(f / 1,5)$, глубина кодирования (разрешение) — $(4 \cdot B)$, количество каналов — 2, время записи не изменилось. Значит, размер файла, полученного при повторной записи:

$$\begin{aligned} ((f / 1,5) \cdot 4 \cdot B \cdot 2 \cdot t) / 2^{23} &= ((f \cdot B \cdot 1 \cdot t) / 2^{23}) \cdot 8 / 1,5 = \\ &= 24 \cdot 8 / 1,5 = 128 \text{ Мб}. \end{aligned}$$

Ответ: 128.

Пример 6.16

Аналоговый звуковой сигнал был записан сначала с использованием 64 уровней дискретизации сигнала, а затем с использованием 4096 уровней дискретизации сигнала. Во сколько раз увеличился информационный объем оцифрованного звука?

- 1) 64
- 2) 8
- 3) 2
- 4) 12

Решение:

$$V(\text{бит}) = f \cdot \log_2 d \cdot k \cdot t,$$

где V — размер файла, f — частота дискретизации, d — количество уровней дискретизации, k — количество каналов, t — время.

$$V_1 = f \cdot \log_2 64 \cdot k \cdot t = f \cdot 6 \cdot k \cdot t;$$

$$V_2 = f \cdot \log_2 4096 \cdot k \cdot t = f \cdot 12 \cdot k \cdot t;$$

$$V_2 / V_1 = 2.$$

Правильный ответ указан под номером 3.

Ответ: 3.

Пример 6.17

Проводилась одноканальная (моно) звукозапись с частотой дискретизации 16 кГц и 24-битным разрешением. В результате был получен файл размером 3 Мбайт, сжатие данных не производилось. Какая из приведенных ниже величин наиболее близка к времени, в течение которого проводилась запись?

- | | |
|---------|----------|
| 1) 30 с | 2) 60 с |
| 3) 90 с | 4) 120 с |

Решение

$$V(\text{бит}) = f(\text{Гц}) \cdot B(\text{бит}) \cdot k \cdot t(\text{с}),$$

где V — размер файла, f — частота дискретизации, B — глубина кодирования, k — количество каналов, t — время.

Значит, время

$$\begin{aligned} t &= V / (f \cdot B \cdot k) = (3 \cdot 2^{23}) / (2^4 \cdot 1000 \cdot 3 \cdot 2^3 \cdot 1) = \\ &= (3 \cdot 2^{23}) / (2^4 \cdot 2^3 \cdot 125 \cdot 3 \cdot 2^3 \cdot 1) = 2^{13} / 125 = \\ &= 65,5 \approx 60 \text{ сек.} \end{aligned}$$

Ответ: 2.

Кодирование изображения

Пример 6.18

Какой минимальный объём памяти (в Кбайт) нужно зарезервировать, чтобы можно было сохранить любое растровое изображение размером 64×64 пикселей при условии, что в изображении могут использоваться 256 различных цветов? В ответе запишите только целое число, единицу измерения писать не нужно.

Решение

$$V(\text{бит}) = X \cdot Y \cdot \log_2 N,$$

где V — объём памяти, X , Y — количество пикселей по горизонтали и вертикали, N — количество цветов.

$$V(\text{Кб}) = (64 \cdot 64 \cdot \log_2 256) / 2^{13} = 2^{12} \cdot 8 / 2^{13} = 4.$$

Ответ: 4.

Пример 6.19

Для хранения растрового изображения размером 64×32 пикселя отвели 1 килобайт памяти. Каково максимально возможное число цветов в палитре изображения?

Решение

$$V(\text{бит}) = X \cdot Y \cdot \log_2 N,$$

где V — объём памяти, X , Y — количество пикселей по горизонтали и вертикали, N — количество цветов.

$$\log_2 N = V / (X \cdot Y) = 2^{13} / (2^6 \cdot 2^5) = 4;$$

$$N = 16.$$

Ответ: 16.

Пример 6.20

Цвет пикселя монитора определяется тремя составляющими: зеленой, синей и красной. Под красную и синюю составляющие одного пикселя отвели по пять бит. Сколько бит отвели под зеленую составляющую одного пикселя, если растровое изображение размером 8×8 пикселей занимает 128 байт памяти?

Решение

Изображение $8 \times 8 = 64$ пикселя занимает 128 байт, значит, один пиксель занимает 2 байта = 16 бит. Под красную и синюю составляющую отвели по 5 бит, значит, под зеленую осталось 6 бит.

Ответ: 6.

Пример 6.21

Цвет пикселя, формируемого принтером, определяется тремя составляющими: голубой, пурпурной и желтой. Под каждую составляющую одного пикселя отвели по четыре бита. В какое количество цветов можно раскрасить пиксель?

Решение

Раз под каждую из трех составляющих отвели по 4 бита, значит, под пиксель отведено всего 12 бит. Количество цветов, в которое можно раскрасить пиксель = $2^{12} = 4096$.

Ответ: 4096.

Пример 6.22

В процессе преобразования растрового графического изображения количество цветов уменьшилось с 65536 до 16. Во сколько раз уменьшится информационный объем графического файла?

Решение

$$V(\text{бит}) = X \cdot Y \cdot \log_2 N,$$

где V — объем памяти, X , Y — количество пикселей по горизонтали и вертикали, N — количество цветов.

$$V_1 = X \cdot Y \cdot \log_2 65536 = X \cdot Y \cdot 16;$$

$$V_2 = X \cdot Y \cdot \log_2 16 = X \cdot Y \cdot 4;$$

$$V_1 / V_2 = 4.$$

Ответ: 4.

Сравнение двух способов передачи данных

Пример 6.23

Документ объемом 5 Мбайт можно передать с одного компьютера на другой двумя способами:

А) Сжать архиватором, передать архив по каналу связи, распаковать.

Б) Передать по каналу связи без использования архиватора.

Какой способ быстрее и насколько, если:

- средняя скорость передачи данных по каналу связи составляет 2^{18} бит в секунду,
- объем сжатого архиватором документа равен 80% от исходного,
- время, требуемое на сжатие документа, — 35 секунд, на распаковку — 3 секунды?

В ответе напишите букву А, если способ А быстрее, или Б, если быстрее способ Б. Сразу после буквы напишите количество секунд, насколько один способ быстрее другого. Так, например, если способ Б быстрее способа А на 23 секунды, в ответе нужно написать Б23. Слов «секунд», «сек.», «с.» к ответу добавлять не нужно.

Решение

Способ А. Общее время складывается из времени сжатия, распаковки и передачи. Время передачи t рассчитывается по формуле $t = V / q$, где V — объем информации, q — скорость передачи данных.

Объем сжатого документа: $5 \cdot 0,8 = 4 \text{ Мб} = 4 \cdot 2^{23} \text{ бит}$.

Найдем общее время:

$$t = 35 \text{ с} + 3 \text{ с} + 4 \cdot 2^{23} \text{ бит} / 2^{18} \text{ бит/с} = 38 + 2^7 \text{ с} = 166 \text{ с}.$$

Способ Б. Общее время совпадает с временем передачи:

$$t = 5 \cdot 2^{23} \text{ бит} / 2^{18} \text{ бит/с} = 5 \cdot 2^5 \text{ с} = 160 \text{ с}.$$

Способ Б быстрее на $166 - 160 = 6 \text{ с}$.

Ответ: Бб.

Определение времени передачи данных**Пример 6.24**

Скорость передачи данных через ADSL-соединение равна 128000 бит/с. Через данное соединение передают файл размером 625 Кбайт. Определите время передачи файла в секундах.

Решение

Время $t = V / q$, где V — объем информации, q — скорость передачи данных.

$$\begin{aligned} t &= 625 \cdot 2^{10} \text{ байт} / (2^7 \cdot 1000) \text{ бит/с} = \\ &= 625 \cdot 2^{13} \text{ бит} / (125 \cdot 2^{10}) \text{ бит/с} = 5 \cdot 2^3 \text{ с} = 40 \text{ с}. \end{aligned}$$

Ответ: 40.

Пример 6.25

Сколько секунд потребуется модему, передающему сообщения со скоростью 28800 бит/с, чтобы передать 100 страниц текста в 30 строк по 60 символов каждая, при условии, что каждый символ кодируется 1 байтом?

Решение

Время $t = V / q$, где V — объем информации, q — скорость передачи данных.

$$t = (100 \cdot 30 \cdot 60 \cdot 8 \text{ бит}) / (28800 \text{ бит/с}) = 50 \text{ с.}$$

Ответ: 50.

Пример 6.26

Сколько секунд потребуется модему, передающему сообщения со скоростью 19200 бит/с, чтобы передать цветное растровое изображение размером 1280 на 800 пикселей, при условии, что цвет каждого пикселя кодируется 24 битами?

Решение

Время $t = V / q$, где V — объем информации, q — скорость передачи данных.

$V = X \cdot Y \cdot I$, где X , Y — количество пикселей по горизонтали и вертикали, I — количество бит, выделяемое под один пиксель.

$$t = (1280 \cdot 800 \cdot 24 \text{ бит}) / (19200 \text{ бит/с}) = 1280 \text{ с.}$$

Ответ: 1280.

Пример 6.27

У Васи есть доступ к Интернет по высокоскоростному одностороннему радиоканалу, обеспечивающему скорость получения им информации 2^{17} бит в секунду. У Пети нет скоростного доступа в Интернет, но есть возможность получать информацию от Васи по низкоскоростному телефонному каналу со средней скоростью 2^{15} бит в секунду. Петя договорился с Васей, что тот будет скачивать для него данные объемом 4 Мбайта по высокоскоростному каналу и ретранслировать их Пете по низкоскоростному каналу. Компьютер Васи может

начать ретрансляцию данных не раньше, чем им будут получены первые 512 Кбайт этих данных. Каков минимально возможный промежуток времени (в секундах), с момента начала скачивания Васей данных, до полного их получения Петей? В ответе укажите только число, слово «секунд» или букву «с» добавлять не нужно.

Решение

Нужно определить, сколько времени будет передаваться файл объемом 4 Мбайта по каналу со скоростью передачи данных 2^{15} бит/с; к этому времени нужно добавить задержку файла у Васи (пока он не получит 512 Кбайт данных по каналу со скоростью 2^{17} бит/с).

Время скачивания данных Петей:

$$t_1 = 4 \cdot 2^{23} \text{ бит} / 2^{15} \text{ бит/с} = 2^{10} \text{ с.}$$

Время задержки:

$$t_2 = 512 \text{ Кбайт} / 2^{17} \text{ бит/с} = 2^{(9+10+3)-17} \text{ с} = 2^5 \text{ с.}$$

Полное время:

$$t_1 + t_2 = 2^{10} \text{ с} + 2^5 \text{ с} = (1024 + 32) \text{ с} = 1056 \text{ с.}$$

Ответ: 1056.

Пример 6.28

Данные объемом 60 Мбайт передаются из пункта А в пункт Б по каналу связи, обеспечивающему скорость передачи данных 2^{19} бит в секунду, а затем из пункта Б в пункт В по каналу связи, обеспечивающему скорость передачи данных 2^{20} бит в секунду. Задержка в пункте Б (время между окончанием приема данных из пункта А и началом передачи в пункт В) составляет 25 секунд. Сколько времени (в секундах) прошло с момента начала передачи данных из пункта А до их полного получения в пункте В? В ответе укажите только число, слово «секунд» или букву «с» добавлять не нужно.

Решение

Полное время складывается из времени передачи из пункта А в пункт Б (t_1), задержки в пункте Б (t_2) и времени передачи из пункта Б в пункт В (t_3).

$$t_1 = (60 \cdot 2^{23}) / 2^{19} = 60 \cdot 16 = 960 \text{ с};$$

$$t_2 = 25 \text{ с};$$

$$t_3 = (60 \cdot 2^{23}) / 2^{20} = 60 \cdot 8 = 480 \text{ с};$$

$$\text{Полное время } t_1 + t_2 + t_3 = 960 + 25 + 480 = 1465 \text{ с.}$$

Ответ: 1465.

Задача № 10

Методы измерения количества информации

Рассмотрим два метода измерения информации:

Алфавитный (технический) метод

При работе с вычислительной техникой *информационным объемом сообщения называют количество двоичных символов, которое используют для кодирования этого сообщения.*

Чтобы найти информационный объем сообщения I , нужно количество символов этого сообщения N умножить на количество бит, выделяемых для кодирования одного символа K : $I = N \cdot K$.

Количество символов в некотором алфавите называется мощностью алфавита.

Несложно понять, что количество слов длиной N , составленных из символов (букв) алфавита мощностью M , равно M^N .

При компьютерном кодировании мощность алфавита равна 2, значит, количество слов длиной N равно 2^N .

Вероятностный метод

Количество информации рассматривается как мера уменьшения неопределенности при получении сообщения.

Количество информации I о событии равно $-\log_2 P$, где P — вероятность этого события:

$$I = -\log_2 P \text{ (формула Шеннона).}$$

Подсчет количества буквенных цепочек

Пример 6.29

Все 5-буквенные слова, составленные из букв А, О, У, записаны в алфавитном порядке. Вот начало списка:

1. ААААА

2. ААААО

3. ААААУ

4. АААОА

.....

Запишите слово, которое стоит на 210-м месте от начала списка.

Решение

Заменим буквы А, О, У на 0, 1, 2 и выпишем начало списка:

1. 00000

2. 00001

3. 00002

4. 00010

...

Полученная запись есть числа, записанные в троичной системе счисления в порядке возрастания. Тогда на 210-м месте будет стоять число 209 (так как первое число 0). Переведем число 209 в троичную систему: $209_{10} = 21202_3$.

Заменим обратно цифры на буквы и получим УОУАУ.

Ответ: УОУАУ.

Пример 6.30

Сколько слов длиной 6 букв, начинающихся с согласной буквы, можно составить из букв Г, О, Д? Каждая буква может входить в слово несколько раз. Слова не обязательно должны быть осмысленными словами русского языка.

Решение

На первом месте может стоять две буквы: Г или Д, на остальных — три буквы.

Слов, начинающихся на Г, 3^5 . Слов, начинающихся на Д, тоже 3^5 . Таким образом, можно составить $2 \cdot 3^5 = 486$ слов.

Ответ: 486.

Пример 6.31

Вася составляет 5-буквенные слова, в которых есть только буквы С, Л, О, Н, причём буква С используется в каждом слове ровно 1 раз. Каждая из других допустимых букв может встречаться в слове любое количество раз или не встречаться совсем. Словом считается любая допустимая последовательность букв, не обязательно осмысленная. Сколько существует таких слов, которые может написать Вася?

Решение

Пусть С стоит в слове на первом месте. Тогда на каждое из оставшихся 4 мест можно поставить независимо одну из 3 букв. То есть всего $3 \cdot 3 \cdot 3 \cdot 3 = 81$ вариант. Таким образом, С можно по очереди поставить на все 5 мест, в каждом случае получая 81 вариант. Итого получается $81 \cdot 5 = 405$ слов.

Ответ: 405.

Пример 6.32

Игорь составляет таблицу кодовых слов для передачи сообщений, каждому сообщению соответствует своё кодовое слово. В качестве кодовых слов Игорь использует 5-буквенные слова, в которых есть только буквы П, И, Р, причем буква П появляется ровно 1 раз. Каждая из других допустимых букв может встречаться в кодовом слове любое количество раз или не встречаться совсем. Сколько различных кодовых слов может использовать Игорь?

Решение

Если буква П стоит на первом месте, тогда на остальные 4 места можно поставить букву И или Р. Таких вариантов $2 \cdot 2 \cdot 2 \cdot 2 = 16$.

Аналогично, поставив П на 2-е место и т.д. до 5-го, получим такое же количество вариантов.

Тогда общее количество различных слов $= 5 \cdot 16 = 80$.

Ответ: 80.

Пример 6.33

Все 6-буквенные слова, составленные из букв С, В, Е, Т, записаны в алфавитном порядке и пронумерованы. Вот начало списка:

1. ВВВВВВ
2. ВВВВВЕ
3. ВВВВВС
4. ВВВВВТ
5. ВВВВЕВ

Под каким номером будет стоять первое начинающееся на Т слово?

Решение

Заменяем буквы В, Е, С, Т на 0, 1, 2, 3 и выпишем начало списка:

1. 000000
2. 000001
3. 000002
4. 000003
5. 000010

...

Полученная запись есть числа, записанные в четверичной системе счисления в порядке возрастания. Первое слово, начинающееся на Т, — 300000. Переведём его в десятичную систему и прибавим 1 (так как под номером 1 стоит 0):

$$300000_4 = 3 \cdot 4^5 = 3072.$$

$$3072 + 1 = 3073.$$

Ответ: 3073.

Пример 6.34

Сколько есть различных символьных последовательностей длины от двух до четырёх в трёхбуквенном алфавите {А, В, С}?

Решение

Если в алфавите M символов, то количество всех возможных слов длиной N равно M^N . Так как длина слова от двух символов до четырех, необходимо сложить количество двух-, трех- и четырехбуквенных слов:

$$3^2 + 3^3 + 3^4 = 117.$$

Ответ: 117.

Количество информации при двоичном (компьютерном) кодировании

Пример 6.35

Объем сообщения — 7,5 Кбайт. Известно, что данное сообщение содержит 7680 символов. Какова мощность алфавита?

Решение

Объем сообщения I , написанного в исходном алфавите мощностью M , содержащего N символов, равен:

$$I = \log_2 M \cdot N;$$

$$I = 7680 \cdot \log_2 M;$$

$$\log_2 M = (7,5 \cdot 2^{13} \text{бит}) / 7680 = (7,5 \cdot 2^{13}) / (15 \cdot 2^9) = 8;$$

$$M = 2^8 = 256.$$

Ответ: 256.

Пример 6.36

Мощность алфавита равна 256. Сколько Кбайт памяти потребуется для сохранения 160 страниц текста, содержащего в среднем 192 символа на каждой странице?

Решение

I (бит) = $N \cdot \log_2 M$, где N — количество символов в сообщении, M — мощность алфавита.

$$I (\text{Кбайт}) = (160 \cdot 192 \cdot \log_2 256) / 2^{13} = (160 \cdot 192 \cdot 8) / 2^{13} = 30.$$

Ответ: 30.

Пример 6.37

Для кодирования нотной записи используется 7 значков-нот. Каждая нота кодируется одним и тем же минимально возможным количеством бит. Чему равен информационный объем в битах сообщения, состоящего из 180 нот?

Решение

I (бит) = $N \cdot \log_2 M$, где N — количество символов в сообщении, M — мощность алфавита.

Если $\log_2 M$ не целое, округляем в большую сторону.

$$M = 7; N = 180.$$

$$I = 180 \cdot \log_2 7 \approx 180 \cdot 3 = 540 \text{ бит.}$$

Ответ: 540.

Количество информации при различных (не компьютерных) способах кодирования

Пример 6.38

Азбука Морзе позволяет кодировать символы для сообщений по радиосвязи, задавая комбинацию точек и тире. Сколько различных символов (цифр, букв, знаков пунктуации и т. д.) можно закодировать, используя код азбуки Морзе длиной не менее четырех и не более пяти сигналов (точек и тире)?

Решение

Мы имеем алфавит из двух букв: точка и тире. Из двух букв можно составить 2^4 четырехбуквенных слова и 2^5 пятибуквенных слов.

Значит, всего можно закодировать $16 + 32 = 48$ различных символов.

Ответ: 48.

Пример 6.39

Некоторый алфавит содержит три различные буквы. Сколько четырехбуквенных слов можно составить из букв данного алфавита (буквы в слове могут повторяться)?

Решение

Количество слов длиной N , составленных из символов (букв) алфавита мощностью M , равно M^N . В нашем случае $N = 4$, $M = 3$. Следовательно, количество слов равно $3^4 = 81$.

Ответ: 81.

Пример 6.40

Световое табло состоит из лампочек. Каждая лампочка может находиться в одном из трех состояний («включено», «выключено» или «мигает»). Какое наименьшее количество лампочек должно находиться на табло, чтобы с его помощью можно было передать 18 различных сигналов?

Решение

Мощность алфавита $M = 3$ («включено», «выключено» или «мигает»).

Количество различных сигналов $18 \leq M^N = 3^N$. (Поскольку равенство не выполняется, N берем с избытком, иначе не сможем закодировать все сигналы.) $N = 3$.

Ответ: 3.

Пример 6.41

На световой панели в ряд расположены 7 лампочек. Каждая из первых двух лампочек может гореть красным, жёлтым или зелёным цветом. Каждая из остальных пяти лампочек может гореть одним из двух цветов — красным или белым. Сколько различных сигналов можно передать с помощью панели (все лампочки должны гореть, порядок цветов имеет значение)?

Решение

При помощи двух первых лампочек можно передать $3^2 = 9$ сигналов. При помощи остальных пяти лампочек можно передать $2^5 = 32$ сигнала. Значит, при помощи всей панели можно передать $9 \cdot 32 = 288$ сигналов.

Ответ: 288.

Вероятностный метод подсчета количества информации

Пример 6.42

В корзине лежат 8 черных шаров и 24 белых. Сколько бит информации несет сообщение о том, что достали черный шар?

Решение

По формуле Шеннона, количество информации I о событии равно $-\log_2 P$, где P — вероятность этого события:
 $I = -\log_2 P$.

Вероятность достать черный шар $P = 8/32 = 1/4$.

$$I = -\log_2 (1/4) = \log_2 4 = 2.$$

Ответ: 2.

Пример 6.43

За четверть Василий Пупкин получил 20 оценок. Сообщение о том, что он вчера получил четверку, несет 2 бита информации. Сколько четверок получил Василий за четверть?

Решение

По формуле Шеннона количество информации I о событии равно $-\log_2 P$, где P — вероятность этого события:
 $I = -\log_2 P$.

Вероятность, что Василий получил четверку, $P = x/20$.

$$2 = -\log_2 (x/20);$$

$$20/x = 4;$$

$$x = 5.$$

Ответ: 5.

Задача № 13

Информационный объем сообщения

Если нам нужно закодировать сообщение, написанное в алфавите мощностью M , при помощи компьютерных кодов, т.е. в алфавите мощностью 2, то для кодирования всех символов исходного алфавита нам нужно под каждый символ выделить такое количество бит K , чтобы выполнялось условие $M \leq 2^K$.

Тогда информационный объем сообщения I , содержащего N символов, будет равен: $I = K \cdot N$.

$I = \log_2 M \cdot N$ (формула Хартли).

Если $\log_2 M$ не является целым числом, округляем значение в большую сторону.

Пример 6.44

В велокроссе участвуют 28 спортсменов. Специальное устройство регистрирует прохождение каждым из участников промежуточного финиша, записывая его номер с использованием минимально возможного количества бит, одинакового для каждого спортсмена. Какой объем памяти будет использован устройством, когда все спортсмены пройдут промежуточный финиш?

- 1) 21 байт
- 2) 140 бит
- 3) 28 бит
- 4) 28 байт

Решение

По формуле Хартли, объем сообщения I , написанного в исходном алфавите мощностью M , содержащего N символов, будет равен: $I = \log_2 M \cdot N$.

Если $\log_2 M$ не является целым числом, округляем значение в большую сторону.



В нашем случае мощность алфавита (все символы, подлежащие кодированию) $M = 28$. N тоже $= 28$, так как все спортсмены прошли финиш.

Тогда $I = \log_2 28 \cdot 28 \approx 5 \cdot 28 = 140$ бит.

Ответ: 2.

Пример 6.45

При регистрации в компьютерной системе каждому пользователю выдается пароль, состоящий из 15 символов и содержащий только символы из 12-символьного набора: $A, B, C, D, E, F, G, H, K, L, M, N$. В базе данных для хранения сведений о каждом пользователе отведено одинаковое и минимально возможное целое число байт. При этом используют посимвольное кодирование паролей, все символы кодируют одинаковым минимально возможным количеством бит. Кроме собственно пароля, для каждого пользователя в системе хранятся дополнительные сведения, для чего выделено целое число байт; это число одно и то же для всех пользователей. Для хранения сведений о 20 пользователях потребовалось 400 байт. Сколько байт выделено для хранения дополнительных сведений об одном пользователе? В ответе запишите только целое число — количество байт.

Решение

Раз для хранения сведений о 20 пользователях потребовалось 400 байт, то для хранения сведений об 1 пользователе нужно 20 байт.

Для хранения пароля, по формуле Хартли, необходимо $\log_2 M \cdot N = \log_2 12 \cdot 15 \approx 4 \cdot 15 = 60$ бит. Но под пароль отведено минимально возможное целое количество байт, значит, под пароль отведено $60/8 \approx 8$ байт (округляем в большую сторону, иначе не получится записать пароль). Значит,

для хранения дополнительных сведений остается $20 - 8 = 12$ байт.

Ответ: 12.

Пример 6.46

В некоторой стране автомобильный номер длиной 8 символов составляют из заглавных букв (задействовано 20 различных букв) и десятичных цифр в любом порядке. Каждый такой номер в компьютерной программе записывается минимально возможным и одинаковым целым количеством байтов (при этом используют посимвольное кодирование и все СИМВОЛЫ кодируются одинаковым и минимально возможным количеством битов). Определите объем памяти, отводимый этой программой для записи 40 номеров.

- 1) 200 байт
- 2) 120 байт
- 3) 320 байт
- 4) 160 байт

Решение

Согласно условию, в номере могут быть использованы 10 цифр (0...9) и 20 букв, всего $10 + 20 = 30$ символов.

По формуле Хартли, для записи номера длиной 8 необходимо $\log_2 M \cdot N = \log_2 30 \cdot 8 \approx 5 \cdot 8 = 40$ бит = 5 байт.

Тогда 40 номеров занимают $5 \cdot 40 = 200$ байт.

Ответ: 1.

Пример 6.47

В некоторой базе данных хранятся записи, содержащие информацию о некоторых датах. Каждая запись содержит три поля: номер года (число от 1 до 2100), номер месяца (число от 1 до 12) и номер дня в месяце (число от 1 до 30). Каждое поле записывается

отдельно от других полей с использованием минимально возможного количества бит. Определите минимальное количество бит, необходимое для кодирования одной записи.

- 1) 19 бит
- 2) 20 бит
- 3) 21 бит
- 4) 22 бита

Решение

Количество бит, необходимое для кодирования M различных значений, равно $\log_2 M$.

Так как поля независимы, то для каждого нужно своё минимальное число бит.

Для поля с номером года необходимо $\log_2 2100 \approx 12$ бит.

Для поля с номером месяца необходимо $\log_2 12 \approx 4$ бит.

Для поля с номером дня необходимо $\log_2 30 \approx 5$ бит.

Итого для одной записи нужно: $12 + 4 + 5 = 21$ бит.

Правильный ответ указан под номером 3.

Ответ: 3.

Задача № 2

Построение и анализ таблиц истинности логических выражений

В компьютере вся информация представлена в двоичной системе счисления, в которой используется две цифры — 0 и 1. Собственно, и цифр как таковых у компьютера нет, а есть электрический сигнал, проходящий по электронным схемам и соединительным проводникам (шинам) компьютера, который может принимать значения «высокий уровень электрического напряжения» (принимаемый нами за 1) и «низкий уровень электрического напряжения» (принимаемый за 0). Для различных действий над этими нулями и единицами нам необходимы специальные операции, которые работают с двоичными переменными. Такие операции называются логическими операциями.

Логические операции и их аргументы принимают только два значения: 1 («истина») и 0 («ложь»).

Таблица истинности выражения определяет его значения при всех возможных комбинациях исходных данных.

Количество строк в таблице истинности выражения от N переменных равно 2^N .

Основные логические операции:

1. *Логическое умножение* (конъюнкция, логическое И). Обозначается: AND, &, $\wedge$.

Таблица истинности:

A	B	A&B
1	1	1
1	0	0
0	1	0
0	0	0

2. *Логическое сложение* (дизъюнкция, логическое ИЛИ).
Обозначается: OR, |, $\vee$.

Таблица истинности:

A	B	A $\vee$ B
1	1	1
1	0	1
0	1	1
0	0	0

3. *Логическое отрицание* (инверсия, логическое НЕ).
Обозначается: NOT, $\neg$, $\bar{A}$.

Таблица истинности:

A	$\neg A$
0	1
1	0

4. *Логическое следование* (импликация). Обозначается: $\rightarrow$.

Таблица истинности:

A	B	A $\rightarrow$ B
1	1	1
1	0	0
0	1	1
0	0	1

5. *Логическое равенство* (эквивалентность). Обозначается: $\leftrightarrow$, $\sim$.

Таблица истинности:

A	B	$A \sim B$
1	1	1
1	0	0
0	1	0
0	0	1

Порядок (приоритет) выполнения логических операций.

Если в выражении нет скобок, то операции выполняются в следующем порядке:

- Логическое отрицание (инверсия, логическое НЕ);
- Логическое умножение (конъюнкция, логическое И);
- Логическое сложение (дизъюнкция, логическое ИЛИ);
- Логическое следование (импликация);
- Логическое равенство (эквивалентность).

Выбор выражения по таблице истинности

Пример 7.1

Дан фрагмент таблицы истинности выражения F :

x_1	x_2	x_3	x_4	x_5	x_6	F
1	1	0	0	0	0	0
1	0	1	0	0	1	0
1	0	0	1	0	0	0

Каким выражением может быть F ?

- 1) $(x_1 \wedge x_2) \vee (x_3 \wedge x_4) \vee (x_5 \wedge x_6)$
- 2) $(x_1 \wedge x_3) \vee (x_3 \wedge x_5) \vee (x_5 \wedge x_1)$
- 3) $(x_2 \wedge x_4) \vee (x_4 \wedge x_6) \vee (x_6 \wedge x_2)$
- 4) $(x_1 \wedge x_4) \vee (x_2 \wedge x_5) \vee (x_3 \wedge x_6)$

Решение

Все представленные варианты ответа — дизъюнкции трёх конъюнкций. Все значения F в таблице равны нулю. Дизъюнкция равна нулю, когда все слагаемые равны нулю.

Рассмотри поочередно все четыре выражения.

1) В первой строке таблицы $x_1 = 1$ и $x_2 = 1$, значит, $x_1 \wedge x_2 = 1$. Выражение не подходит.

2) Во второй строке таблицы $x_1 = 1$ и $x_3 = 1$, значит, $x_1 \wedge x_3 = 1$. Выражение не подходит.

3) Подставим в третье выражение поочередно значения всех строк таблицы:

Первая строка

$$(x_2 \wedge x_4) \vee (x_4 \wedge x_6) \vee (x_6 \wedge x_2) = \\ = (1 \wedge 0) \vee (0 \wedge 0) \vee (0 \wedge 1) = 0 \vee 0 \vee 0 = 0.$$

Вторая строка

$$(x_2 \wedge x_4) \vee (x_4 \wedge x_6) \vee (x_6 \wedge x_2) = \\ = (0 \wedge 0) \vee (0 \wedge 1) \vee (1 \wedge 0) = 0 \vee 0 \vee 0 = 0.$$

Третья строка

$$(x_2 \wedge x_4) \vee (x_4 \wedge x_6) \vee (x_6 \wedge x_2) = \\ = (0 \wedge 1) \vee (1 \wedge 0) \vee (0 \wedge 0) = 0 \vee 0 \vee 0 = 0.$$

Выражение подходит.

4) В третьей строке таблицы $x_1 = 1$ и $x_4 = 1$, значит, $x_1 \wedge x_4 = 1$. Выражение не подходит.

Ответ: 3.

Пример 7.2

Для таблицы истинности функции F известны значения только некоторых ячеек:

x_1	x_2	x_3	x_4	x_5	x_6	x_7	F
			1		0		1
			0			0	1
0			1				0

Каким выражением может быть F ?

$$1) x1 \wedge x2 \wedge x3 \wedge \neg x4 \wedge x5 \wedge x6 \wedge \neg x7$$

$$2) x1 \vee \neg x2 \vee x3 \vee \neg x4 \vee \neg x5 \vee x6 \vee \neg x7$$

$$3) \neg x1 \wedge x2 \wedge \neg x3 \wedge x4 \wedge x5 \wedge x6 \wedge x7$$

$$4) x1 \vee x2 \vee \neg x3 \vee x4 \vee x5 \vee \neg x6 \vee x7$$

Решение

Рассмотри поочередно все четыре выражения.

Выражение является конъюнкцией переменных и их отрицаний. Конъюнкция равна единице, когда все операнды равны единице. В первой строке $x6 = 0$, а значит, и все выражение F равно нулю, что не соответствует таблице истинности.

Выражение является дизъюнкцией переменных и их отрицаний. Дизъюнкция равна единице, когда хотя бы один операнд равен единице. Подставим во второе выражение поочередно значения всех строк таблицы:

Первая строка

$$x1 \vee \neg x2 \vee x3 \vee \neg x4 \vee \neg x5 \vee x6 \vee \neg x7 =$$

$= x1 \vee \neg x2 \vee x3 \vee 0 \vee \neg x5 \vee 0 \vee \neg x7$ может принимать значение 1, если хотя бы один из операндов равен 1.

Вторая строка

$$x1 \vee \neg x2 \vee x3 \vee \neg x4 \vee \neg x5 \vee x6 \vee \neg x7 =$$

$$= x1 \vee \neg x2 \vee x3 \vee 1 \vee \neg x5 \vee x6 \vee 1 = 1$$

Третья строка

$$x1 \vee \neg x2 \vee x3 \vee \neg x4 \vee \neg x5 \vee x6 \vee \neg x7 =$$

$= 0 \vee \neg x2 \vee x3 \vee 0 \vee \neg x5 \vee x6 \vee \neg x7$ может принимать значение 0, если все остальные операнды равны 0.

Выражение является конъюнкцией переменных и их отрицаний. Конъюнкция равна единице, когда все операнды равны единице. Во второй строке $x4 = 0$, а значит, и все выражение F равно нулю, что не соответствует таблице истинности.

Выражение является дизъюнкцией переменных и их отрицаний. Дизъюнкция равна единице, когда хотя бы один операнд равен единице. В третьей строке $x_4 = 1$, значит, и все выражение F равно 1, что не соответствует таблице истинности.

Ответ: 2.

Пример 7.3

Логическая функция F задаётся выражением $(\neg z) \wedge x \vee x \wedge y$. Определите, какому столбцу таблицы истинности функции F соответствует каждая из переменных x, y, z .

Перем. 1	Перем. 2	Перем. 3	Функция
???	???	???	F
0	0	0	0
0	0	1	1
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	0
1	1	0	0
1	1	1	1

В ответе напишите буквы x, y, z в том порядке, в котором идут соответствующие им столбцы (сначала — буква, соответствующая 1-му столбцу; затем — буква, соответствующая 2-му столбцу; затем — буква, соответствующая 3-му столбцу). Буквы в ответе пишите подряд, никаких разделителей между буквами ставить не нужно.

Пример. Пусть задано выражение $x \rightarrow y$, зависящее от двух переменных x и y , и таблица истинности:

Перем. 1	Перем. 2	Функция
???	???	F
0	0	1
0	1	0
1	0	1
1	1	1

Тогда 1-му столбцу соответствует переменная y , а 2-му столбцу соответствует переменная x . В ответе нужно написать: yx .

Решение

Выражение $(\neg z) \wedge x \vee x \wedge y$ является дизъюнкцией двух конъюнкций: $((\neg z) \wedge x) \vee (x \wedge y)$. В обеих конъюнкциях присутствует x . То есть при $x = 0$ все выражение равно 0. Это выполняется только при Перем. 3 = x .

Выражение равно 1, если $x = 1$ и выполняется хотя бы одно из условий: $y = 1$ или $z = 0$. Из четвертой строки следует, что Перем.1 = z , а Перем.2 = y .

Ответ: zyx .

Подсчет количества наборов значений переменных

Пример 7.4

Дано логическое выражение, зависящее от 6 логических переменных:

$$x_1 \wedge \neg x_2 \wedge x_3 \wedge \neg x_4 \wedge x_5 \wedge x_6.$$

Сколько существует различных наборов значений переменных, при которых выражение истинно?

- 1) 1 2) 2 3) 63 4) 64

Решение

Выражение является конъюнкцией шести операндов. Конъюнкция равна 1 (выражение истинно), когда все опе-

ранды равны 1. То есть существует единственный набор значений переменных.

Ответ: 1.

Пример 7.5

Дан фрагмент таблицы истинности для выражения F :

x_1	x_2	x_3	x_4	x_5	x_6	F
0	0	1	0	0	0	0
1	0	1	0	1	1	1
0	1	1	1	0	0	1

Укажите минимально возможное число различных строк полной таблицы истинности этого выражения, в которых значение x_1 совпадает с F .

Решение

Полная таблица истинности выражения с 6-ю переменными содержит $2^6 = 64$ строки. В приведенном фрагменте таблицы x_1 совпадает с F в двух строках. Во всех остальных может не совпадать. Тогда минимально возможное количество строк, где значения x_1 и F совпадают, будет 2.

Ответ: 2.

Пример 7.6

Дан фрагмент таблицы истинности для выражения F :

x_1	x_2	x_3	x_4	x_5	x_6	x_7	F
0	0	1	1	0	0	1	0
0	1	0	0	1	1	0	1
0	0	0	0	1	1	1	1
1	0	1	0	1	1	0	1
0	1	1	1	0	1	0	1

Укажите максимально возможное число различных строк полной таблицы истинности этого выражения, в которых значение x_4 не совпадает с F .

Решение

Полная таблица истинности выражения с 7-ю переменными содержит $2^7 = 128$ строк. В приведенном фрагменте таблицы x_4 не совпадает с F в 4-х строках, а совпадает в одной строке. Во всех остальных может не совпадать. Тогда максимально возможное количество строк, где значения x_4 и F не совпадают, будет 127.

Ответ: 127.

Пример 7.7

Каждое из логических выражений A и B зависит от одного и того же набора из 5 переменных. В таблицах истинности обоих выражений в столбцах значений стоит ровно по 17 единиц в каждой таблице. Каково максимально возможное число единиц в столбце значений таблицы истинности выражения $\neg (A \wedge B)$?

Решение

Раз каждое из логических выражений A и B зависит от 5 переменных, значит, в таблицах истинности этих выражений по $2^5 = 32$ строк. Из них по 17 единиц и по 15 нулей. Максимальное число единиц в столбце значений таблицы истинности выражения $\neg (A \wedge B)$ будет, когда максимально количество нулей в столбце значений таблицы истинности выражения $A \wedge B$. А это достигается, когда единице в столбце A соответствует ноль в столбце B , и наоборот. Так как нулей в каждой таблице по 15, то максимальное количество наборов $(0, 1)$ и $(1, 0)$ для выражений A и B равно 30.

Ответ: 30.

Задача № 18

Основные понятия и законы математической логики

Законы алгебры логики

	Для И	Для ИЛИ
двойного отрицания	$\neg\neg(A) = A$	
исключения третьего	$A \& \neg A = 0$	$A \vee \neg A = 1$
исключения констант	$A \& 1 = A; A \& 0 = 0$	$A \vee 0 = A; A \vee 1 = 1$
повторения	$A \& A = A$	$A \vee A = A$
поглощения	$A \& (A \vee B) = A$	$A \vee A \& B = A$
переместительный	$A \& B = B \& A$	$A \vee B = B \vee A$
сочетательный	$A \& (B \& C) = (A \& B) \& C$	$A \vee (B \vee C) = (A \vee B) \vee C$
распределительный	$A \vee B \& C = (A \vee B) \& (A \vee C)$	$A \& (B \vee C) = A \& B \vee A \& C$
де Моргана	$\neg(A \& B) = \neg A \vee \neg B$	$\neg(A \vee B) = \neg A \& \neg B$

Преобразование логического выражения

Пример 7.8

Какое логическое выражение равносильно выражению $\neg(A \vee \neg B)$?

1) $A \vee B$

2) $A \wedge B$

3) $\neg A \vee \neg B$

4) $\neg A \wedge B$

Решение

Применим формулу де Моргана, а затем формулу двойного отрицания:

$$\neg(A \vee \neg B) = \neg A \wedge \neg(\neg B) = \neg A \wedge B.$$

Ответ: 4.

Поиск слова, удовлетворяющего условию логического высказывания

Пример 7.9

Для какого имени истинно высказывание:

*(Вторая буква гласная $\rightarrow$ Первая буква гласная)
Последняя буква согласная?*

- | | |
|----------|-----------|
| 1) ИРИНА | 2) МАКСИМ |
| 3) МАРИЯ | 4) СТЕПАН |

Решение

Высказывание является конъюнкцией двух выражений *(Вторая буква гласная $\rightarrow$ Первая буква гласная)* и *Последняя буква согласная*. Конъюнкция истинна тогда, когда все операнды истинны. Значит, выражение *Последняя буква согласная* должно быть истинным. Этому условию удовлетворяют имена под номерами 2 и 4.

Поочередно подставим в высказывание значения выражений для имен 2 и 4:

2) МАКСИМ

Вторая буква гласная = 1

Первая буква гласная = 0

Последняя буква согласная = 1

$(1 \rightarrow 0) \wedge 1 = 0$. Высказывание ложно.

4) СТЕПАН

Вторая буква гласная = 0

Первая буква гласная = 0

Последняя буква согласная = 1

$(0 \rightarrow 0) \wedge 1 = 1 \wedge 1 = 0$. Высказывание истинно.

Ответ: 4.

Пример 7.10

Для какого названия животного ложно высказывание:
Четвертая буква гласная $\rightarrow \neg$ (*Вторая буква согласная*)?

- | | |
|------------|-----------|
| 1) Собака | 2) Жираф |
| 3) Верблюд | 4) Страус |

Решение

Высказывание является импликацией выражений *Четвертая буква гласная* и $\neg$ (*Вторая буква согласная*). Импликация ложна только тогда, когда первый операнд равен 1 (истина), а второй — 0 (ложь). То есть выражение *Четвертая буква гласная* должно быть истинным. Этому условию удовлетворяют названия животных под номерами 1, 2, 4.

Выражение $\neg$ (*Вторая буква согласная*) должно быть ложно, т.е. выражение *Вторая буква согласная* должно быть истинным. Из выбранных названий этому требованию удовлетворяет только название под номером 4 (Страус).

Ответ: 4.

**Поиск числа, удовлетворяющего условию
 логического высказывания**

Пример 7.11

Для какого из приведённых чисел X истинно логическое условие:

$\neg((X \text{ кратно } 5) \rightarrow (X \text{ кратно } 25))$?

- 1) 37
- 2) 59
- 3) 65
- 4) 125

Решение

Для того чтобы логическое условие $\neg((X \text{ кратно } 5) \rightarrow (X \text{ кратно } 25))$ было истинным, необходимо, чтобы условие $(X \text{ кратно } 5) \rightarrow (X \text{ кратно } 25)$ было ложным. Импликация возвращает ложь, только если первый операнд равен 1 (истина), а второй — 0 (ложь).

То есть число X должно быть кратно 5, но не кратно 25.

Этому условию удовлетворяет только число под номером 3 (65).

Ответ: 3.

Пример 7.12

Для какого из значений числа Y высказывание $(Y < 5) \wedge (Y > 1) \rightarrow (Y > 5)$ будет истинным?

1) 1 2) 2 3) 3 4) 4.

Решение

Высказывание является конъюнкцией выражений $(Y < 5) \wedge ((Y > 1) \rightarrow (Y > 5))$. Конъюнкция истинна, когда все операнды истинны. Значит, выражение $(Y < 5)$ должно быть истинно. Этому условию удовлетворяют все предложенные числа.

Выражение $(Y > 1) \rightarrow (Y > 5)$ тоже должно быть истинно. Импликация возвращает ложь, только если первый операнд равен 1 (истина), а второй — 0 (ложь). То есть отбросим значения, которые больше 1, но не больше 5. Останется только значение под номером 1 (1).

Ответ: 1.

Пример 7.13

Для какого числа X истинно высказывание $(X \cdot (X - 8) > -25 + 2 \cdot X)$ и $(X > 7)$.

1) 4 2) 5 3) 6 4) 7

Решение

Импликация возвращает ложь, только если первый операнд равен 1 (истина), а второй — 0 (ложь). Т.е. отбросим значения, для которых $X \cdot (X - 8) > -25 + 2 \cdot X$ и $X \leq 7$.

Рассмотрим систему неравенств

$$\begin{cases} X^2 - 10X + 25 > 0; \\ X \leq 7. \end{cases}$$

$$\begin{cases} (X - 5)^2 > 0; \\ X \leq 7. \end{cases}$$

Подходят все предложенные числа, кроме 5 (число под номером 2). Это и есть ответ.

Ответ: 2.

Пример 7.14

Обозначим через $m \& n$ поразрядную конъюнкцию неотрицательных целых чисел m и n . Так, например, $14 \& 5 = 1110_2 \& 0101_2 = 0100_2 = 4$. Для какого наименьшего неотрицательного целого числа A формула $x \& 25 \neq 0 \rightarrow (x \& 17 = 0 \rightarrow x \& A \neq 0)$ тождественно истинна (т.е. принимает значение 1 при любом неотрицательном целом значении переменной x)?

Решение

Для наглядности введем обозначения: $A \equiv (x \& A \neq 0)$; $B \equiv (x \& 25 \neq 0)$; $C \equiv (x \& 17 = 0)$.

Тогда формула принимает вид: $B \rightarrow (C \rightarrow A) = 1$.

Заменяем первую импликацию: $\neg B \vee (C \rightarrow A) = 1$.

Заменяем импликацию в скобках: $\neg B \vee (\neg C \vee A) = 1$.

В результате имеем: $\neg B \vee \neg C \vee A = 1$.

$x \& 25 = 0 \vee x \& 17 \neq 0 \vee x \& A \neq 0 = 1$.

Выражение является дизъюнкцией трех операндов. Дизъюнкция истинна, когда хотя бы один операнд принимает значение «истина» (1).

$25_{10} = 11001_2$, тогда $x \& 25 = 0$ истинно для всех x , имеющих нули в 0-м, 3-м и 4-м (справа) разрядах двоичной записи: $x = *...*00**0$

$17_{10} = 10001_2$, тогда $x \& 17 \neq 0$ истинно для всех x , имеющих единицы в 0-м или 4-м разряде: $x = *...*1$ или $x = *...1****$.

«Незакрытыми» (не входящими ни в первое, ни во второе множество) на числовой оси остались x , имеющие нули в 0-м и 4-м разрядах и единицу в 3-м разряде: $x = *...*01**0$.

Значит, A должно быть таким, чтобы конъюнкция с оставшимися числами x не была равна нулю, т.е. в 3-м разряде двоичной записи числа A должна стоять единица. Наименьшим таким числом является $1000_2 = 8_{10}$.

Ответ: 8.

Поиск числового отрезка, удовлетворяющего условию логического высказывания

Пример 7.15

На числовой прямой даны два отрезка: $P = [3, 13]$ и $Q = [7; 17]$. Выберите такой отрезок A , чтобы формула $((x \in A) \rightarrow (x \in P)) \vee \neg (x \in Q)$

была тождественно истинна, т.е. принимала значение 1 при любом значении переменной x .

- 1) $[5; 20]$
- 2) $[10; 25]$
- 3) $[15; 30]$
- 4) $[20; 35]$

Решение

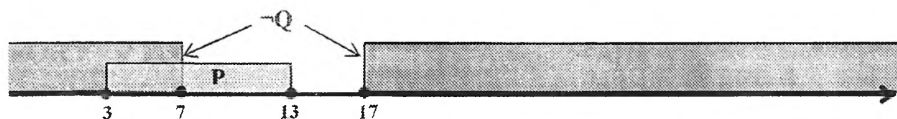
Введем обозначения:

$(x \in A) \equiv A; (x \in P) \equiv P; (x \in Q) \equiv Q$.

Применив преобразование импликации, получаем:

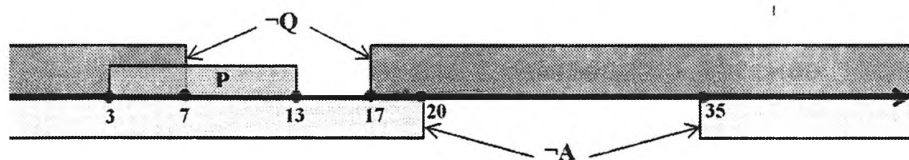
$$\neg A \vee P \vee \neg Q.$$

Изобразим множества P и $\neg Q$ на числовой прямой:



Выражение должно быть истинно для любого x , значит нужно «закрасить» всю числовую прямую. Для этого выражение $\neg A$ должно «закрасить» оставшийся отрезок $[13; 17]$, т.е. быть истинным на этом отрезке. Тогда выражение A должно быть истинно внутри промежутка, который не имеет ни одной общей точки с отрезком $[13; 17]$.

Из всех отрезков только отрезок $[20, 35]$ удовлетворяет этим условиям:



Правильный ответ указан под номером 4.

Ответ: 4

Пример 7.16

На числовой прямой даны три отрезка: $P = [15, 30]$, $Q = [5, 10]$ и $R = [10, 20]$. Выберите такой отрезок A , на котором формула

$$((x \in P) \rightarrow (x \in Q)) \wedge (x \notin A) \wedge (x \in R)$$

тождественно ложна, то есть принимает значение 0 при любом значении переменной x .

- | | |
|---------------|---------------|
| 1) $[0; 12]$ | 2) $[10; 17]$ |
| 3) $[15; 20]$ | 4) $[15; 30]$ |

Решение

Введем обозначения:

$$(x \in A) \equiv A; (x \in P) \equiv P; (x \in Q) \equiv Q; (x \in R) \equiv R.$$

Тогда формула принимает вид:

$$(P \rightarrow Q) \wedge \neg A \wedge R = 0.$$

Применив преобразование импликации, получаем:

$$(\neg P \vee Q) \wedge \neg A \wedge R = 0.$$

Анализировать истинное высказывание удобнее, поэтому применим отрицание к исходному ложному высказыванию и получим истинное:

$$\neg ((\neg P \vee Q) \wedge \neg A \wedge R) = 1.$$

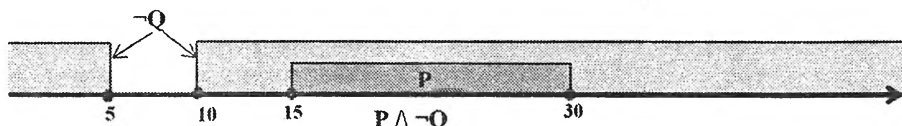
Последовательно преобразуем формулу:

$$\neg(\neg P \vee Q) \vee A \vee \neg R = 1;$$

$$(P \wedge \neg Q) \vee A \vee \neg R = 1.$$

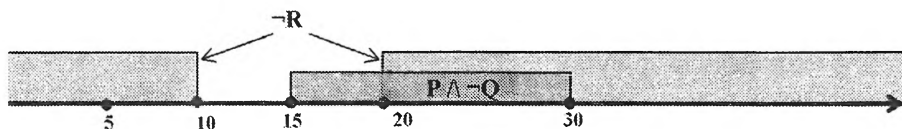
Изобразим множества на числовой прямой:

$(P \wedge \neg Q)$ — это пересечение множеств P и $\neg Q$:



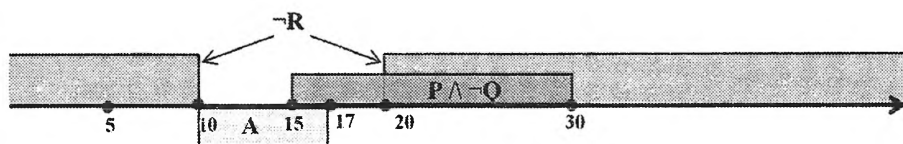
В нашем случае $(P \wedge \neg Q)$ совпадает с P и является отрезком $[15; 30]$.

Тогда $(P \wedge \neg Q) \vee \neg R$ закрывает числовую ось, за исключением отрезка $[10, 15]$:



Выражение $(P \wedge \neg Q) \vee A \vee \neg R$ должно быть истинно для любого x , значит, нужно «закрасить» всю числовую прямую. Для этого выражение A должно «закрасить» оставшийся отрезок $[10; 15]$, т.е. быть истинным на этом отрезке.

Из всех отрезков только отрезок $[10; 17]$ удовлетворяет этим условиям:



Ответ: 2.

Пример 7.17

На числовой прямой даны три отрезка: $P = [5; 10]$, $Q = [15; 20]$ и $R = [25; 30]$. Выберите такой интервал A , на котором формулы

$$(x \in P) \rightarrow (x \in P) \text{ и } (x \in Q) \rightarrow (x \notin R)$$

тождественно равны, то есть принимают равные значения при любом значении переменной x (за исключением, возможно, конечного числа точек).

1) $[5; 10]$

2) $[15; 20]$

3) $[10; 20]$

4) $[15; 25]$

Решение

Введем обозначения:

$$(x \in A) \equiv A; (x \in P) \equiv P; (x \in Q) \equiv Q; (x \in R) \equiv R.$$

Тогда формулы примут вид:

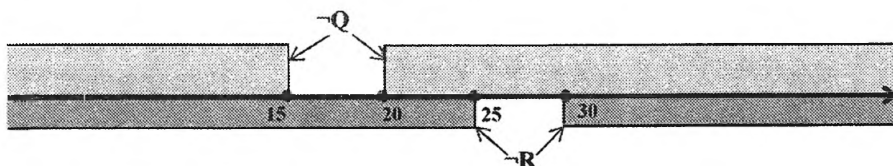
$$(A \rightarrow P) \text{ и } (Q \rightarrow \neg R).$$

Заменив импликацию, получим:

$$(\neg A \vee P) \text{ и } (\neg Q \vee \neg R).$$

Формулы должны быть тождественно равны, значит, их изображения на числовой прямой должны быть одинаковы (за исключением, возможно, конечного числа точек).

Изобразим на числовой прямой выражение $(\neg Q \vee \neg R)$:



Числовая прямая полностью закрашена, т.е. выражение истинно для любых значений x . Значит, выражение $(\neg A \vee P)$ тоже должно быть истинным на всей числовой прямой. Множество P — это отрезок $[5, 10]$, значит выражение $\neg A$ должно «закрасить» оставшуюся часть числовой оси, т.е. быть истинным на этом промежутке. Тогда, выражение A должно быть истинно внутри промежутка $[5; 10]$.

Из всех отрезков только отрезок $[5; 10]$ удовлетворяет этим условиям:



Ответ: 1

Пример 7.18

На числовой прямой даны два отрезка: $P = [25; 50]$ и $Q = [32; 47]$. Укажите наибольшую возможную длину промежутка A , для которого формула

$$(\neg(x \in A) \rightarrow (x \in P)) \rightarrow ((x \in A) \rightarrow (x \in Q))$$

тождественно истинна, то есть принимает значение 1 при любом значении переменной x .

Решение:

Введем обозначения:

$$(x \in A) \equiv A; (x \in P) \equiv P; (x \in Q) \equiv Q.$$

Тогда формула примет вид:

$$(\neg A \rightarrow P) \rightarrow (A \rightarrow Q)$$

Преобразуем данное выражение (заменяем импликацию):

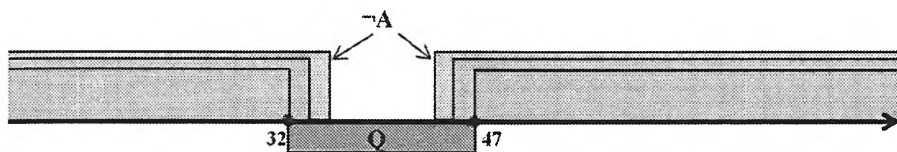
$$(A \vee \neg P) \rightarrow (\neg A \vee Q)$$

$$\neg (A \vee \neg P) \vee (\neg A \vee Q)$$

$$(\neg A \wedge P) \vee \neg A \vee Q$$

$$((\neg A \wedge P) \vee \neg A) \vee Q$$

$$\neg A \vee Q$$



Выражение $(\neg A \vee Q)$ должно быть истинным на всей числовой прямой. Множество Q — это отрезок $[32; 47]$, значит, выражение $\neg A$ должно «закрасить» оставшуюся часть числовой оси, т.е. быть истинным на этом промежутке. Тогда выражение A должно быть истинно внутри промежутка $[32; 47]$. Тогда максимальная длина отрезка A достигается, когда A совпадает с Q и равна 15.

Ответ: 15.

Поиск множества чисел, удовлетворяющего условию логического высказывания

Пример 7.19

Элементами множеств A , P , Q являются натуральные числа, причем

$$P = \{2, 4, 6, 8, 10, 12, 14, 16, 18, 20\},$$

$$Q = \{3, 6, 9, 12, 15, 18, 21, 24, 27, 30\}.$$

Известно, что выражение

$$((x \in A) \rightarrow (x \in P)) \vee (\neg(x \in Q) \rightarrow \neg(x \in A))$$

истинно (т.е. принимает значение 1) при любом значении переменной x .

Определите наибольшее возможное количество элементов в множестве A .

Решение

Введем обозначения:

$$(x \in A) \equiv A; (x \in P) \equiv P; (x \in Q) \equiv Q.$$

Тогда выражение примет вид:

$$(A \rightarrow P) \vee (\neg Q \rightarrow \neg A).$$

Преобразуем выражение (заменяем импликацию):

$$(\neg A \vee P) \vee (Q \vee \neg A)$$

$$\neg A \vee P \vee Q$$

Чтобы выражение было истинно при любом значении переменной x , все натуральные числа должны либо входить в P , либо входить в Q , либо не входить в A . То есть $\neg A$ — это все числа, не входящие ни в P , ни в Q . Значит A — это числа, входящие в P или Q . Наибольшее возможное количество элементов в множестве A — это количество всех различных элементов множеств P и Q . Таких элементов 17.

Ответ: 17.

Пример 7.20

Элементами множества A являются натуральные числа. Известно, что выражение

$$(x \in \{2, 4, 6, 8, 10, 12\}) \rightarrow (((x \in \{3, 6, 9, 12, 15\}) \wedge \neg(x \in A)) \rightarrow \neg(x \in \{2, 4, 6, 8, 10, 12\}))$$

истинно (т.е. принимает значение 1) при любом значении переменной x . Определите наименьшее возможное значение суммы элементов множества A .

Решение

Введем обозначения:

$$(x \in \{2, 4, 6, 8, 10, 12\}) \equiv P;$$

$$(x \in \{3, 6, 9, 12, 15\}) \equiv Q; (x \in A) \equiv A.$$

Тогда выражение примет вид:

$$P \rightarrow ((Q \wedge \neg A) \rightarrow \neg P).$$

Преобразуем выражение (заменяем импликацию):

$$P \rightarrow (\neg(Q \wedge \neg A) \vee \neg P);$$

$$\neg P \vee (\neg(Q \wedge \neg A) \vee \neg P);$$

$$\neg P \vee \neg Q \vee A.$$

Выражение $\neg P \vee \neg Q$ истинно при всех значениях x , кроме значений 6 и 12. Следовательно, промежуток A должны содержать точки 6 и 12. То есть минимальный набор точек в

промежутке $A \equiv \{6; 12\}$. Сумма элементов множества A равна 18.

Ответ: 18.

Пример 7.21

Элементами множества A являются натуральные числа. Известно, что выражение

$$((x \in \{3, 5, 7, 11, 12, 15\}) \rightarrow (x \in \{5, 6, 12, 15\})) \vee (x \in A)$$

истинно (т.е. принимает значение 1) при любом значении переменной x .

Определите наименьшее возможное значение произведения элементов множества A .

Решение

Введем обозначения:

$$(x \in \{3, 5, 7, 11, 12, 15\}) \equiv P;$$

$$(x \in \{5, 6, 12, 15\}) \equiv Q; (x \in A) \equiv A.$$

Тогда выражение примет вид:

$$(P \rightarrow Q) \vee A.$$

Преобразуем выражение (заменяем импликацию):

$$\neg P \vee Q \vee A.$$

Выражение $\neg P \vee Q$ истинно при всех значениях x , кроме значений 3, 7, 11. Следовательно, промежуток A должен содержать эти значения. То есть минимальный набор точек в промежутке $A \equiv \{3, 7, 11\}$. Произведение элементов множества A равно 231.

Ответ: 231.

Задача № 23

Построение и преобразование логических выражений

Решение логических уравнений

Пример 7.22

Сколько различных решений имеет уравнение:

$$\neg((J \rightarrow K) \rightarrow (L \wedge M \wedge N)) \vee \neg((L \wedge M \wedge N) \rightarrow \rightarrow (\neg J \vee K)) \vee (M \wedge J) = 0.$$

Решение

Выражение является дизъюнкцией трех операндов.

Поочередно преобразуем два первых операнда дизъюнкции:

$$\begin{aligned} \neg((J \rightarrow K) \rightarrow (L \wedge M \wedge N)) &= \neg((\neg J \vee K) \rightarrow (L \wedge M \wedge N)) = \\ &= \neg(\neg(\neg J \vee K) \vee (L \wedge M \wedge N)) = \neg((J \wedge \neg K) \vee (L \wedge M \wedge N)) = \\ &= (\neg J \vee K) \wedge (L \wedge M \wedge N). \end{aligned}$$

$$\begin{aligned} \neg((L \wedge M \wedge N) \rightarrow (\neg J \vee K)) &= \neg(\neg(L \wedge M \wedge N) \vee (\neg J \vee K)) = \\ &= (L \wedge M \wedge N) \wedge \neg(\neg J \vee K) = L \wedge M \wedge N \wedge J \wedge \neg K. \end{aligned}$$

Исходное уравнение примет вид:

$$\underbrace{((\neg J \vee K) \wedge (\neg L \vee \neg M \vee \neg N))}_{\text{операнд 1}} \vee \underbrace{(L \wedge M \wedge N \wedge J \wedge \neg K)}_{\text{операнд 2}} \vee \underbrace{(M \wedge J)}_{\text{операнд 3}} = 0.$$

Дизъюнкция равна нулю, если все операнды равны нулю.

Рассмотрим третий операнд дизъюнкции:

$$M \wedge J = 0.$$

Он равен нулю на наборах:

J	M
0	0
0	1
1	0

Второй операнд дизъюнкции ($L \wedge M \wedge N \wedge J \wedge \neg K$) равен нулю на тех же наборах J, M при любых значениях K, L, N (так как конъюнкция равна нулю, если один из операндов равен нулю).

Рассмотрим значения третьего операнда дизъюнкции $((\neg J \vee K) \wedge (\neg L \vee \neg M \vee \neg N))$.

При $J = 0$ и $M = 0$ третий операнд принимает значение $(1 \vee K) \wedge (\neg L \vee 1 \vee \neg N) = 1$ независимо от значений K, L, N . Но по условию он равен нулю, значит на этом наборе J, M решений нет.

При $J = 0$ и $M = 1$ третий операнд принимает значение $(1 \vee K) \wedge (\neg L \vee 0 \vee \neg N) = 1 \wedge (\neg L \vee 0 \vee \neg N) = (\neg L \vee 0 \vee \neg N)$. Равенство нулю достигается, когда $L = 1$ и $N = 1$. При любом значении K (2 набора):

J	M	K	L	N
0	1	0	1	1
0	1	1	1	1

При $J = 1$ и $M = 0$ третий операнд принимает значение $(0 \vee K) \wedge (\neg L \vee 1 \vee \neg N) = (0 \vee K) \wedge 1 = (0 \vee K)$. Равенство нулю достигается, когда $K = 0$. При любых значениях L, N (4 набора):

J	M	K	L	N
1	0	0	0	0
1	0	0	0	1
1	0	0	1	0
1	0	0	1	1

Итого, $2 + 4 = 6$ наборов.

Ответ: 6.

Пример 7.23

Каково наибольшее целое положительное число X , при котором ложно высказывание:

$$(X(X + 1) > 55) \rightarrow (X \cdot X > 50)?$$

Решение

Импликация ложна только тогда, когда первый операнд принимает значение «истина», а второй — «ложь». То есть имеем систему неравенств:

$$\begin{cases} X(X + 1) > 55; \\ X \cdot X < 50. \end{cases}$$

Поскольку требуется найти наибольшее целое X , нет необходимости решать квадратное неравенство, а простым подбором видим, что первое неравенство выполняется при $X \geq 7$ ($7 \cdot 8 = 56$), а второе — при $X \leq 7$ ($7 \cdot 7 = 49$). Тогда общее решение двух неравенств $X = 7$.

Ответ: 7.

Решение систем логических уравнений методом замены переменных

Метод замены переменных применяется, если некоторые переменные входят в состав уравнений только в виде конкретного выражения, и никак иначе. Тогда это выражение можно обозначить новой переменной.

Пример 7.24

Сколько существует различных наборов значений логических переменных $x_1, x_2, x_3, x_4, x_5, x_6, x_7, x_8$, которые удовлетворяют всем перечисленным ниже условиям?

$$(x_1 \rightarrow x_2) \rightarrow (x_3 \rightarrow x_4) = 1$$

$$(x_3 \rightarrow x_4) \rightarrow (x_5 \rightarrow x_6) = 1$$

$$(x_5 \rightarrow x_6) \rightarrow (x_7 \rightarrow x_8) = 1$$

В ответе не нужно перечислять все различные наборы значений переменных $x_1, x_2, x_3, x_4, x_5, x_6, x_7, x_8$, при которых выполнена данная система равенств. В качестве ответа вам нужно указать количество таких наборов.

Решение

Сделаем замену переменных:

$$(x_1 \rightarrow x_2) = y_1; (x_3 \rightarrow x_4) = y_2;$$

$$(x_5 \rightarrow x_6) = y_3; (x_7 \rightarrow x_8) = y_4.$$

Тогда можно записать систему в виде одного уравнения:

$(y_1 \rightarrow y_2) \wedge (y_2 \rightarrow y_3) \wedge (y_3 \rightarrow y_4) = 1$. Конъюнкция равна 1 (истинна), когда каждый операнд принимает значение 1. То есть каждая из импликаций должна быть истинна, а это выполняется при всех значениях, кроме $(1 \rightarrow 0)$. То есть в таблице значений переменных y_1, y_2, y_3, y_4 единица не должна стоять левее нуля:

y_1	y_2	y_3	y_4
0	0	0	0
0	0	0	1
0	0	1	1
0	1	1	1
1	1	1	1

То есть условия выполняются для 5 наборов y_1 – y_4 .

Так как $y_1 = x_1 \rightarrow x_2$, то значение $y_1 = 0$ достигается на единственном наборе x_1, x_2 : $(1, 0)$, а значение $y_1 = 1$ — на трех наборах x_1, x_2 : $(0, 0)$, $(0, 1)$, $(1, 1)$. Аналогично для y_2, y_3, y_4 .

Поскольку каждый набор (x_1, x_2) для переменной y_1 сочетается с каждым набором (x_3, x_4) для переменной y_2 и т.д., то количества наборов переменных x перемножаются:

y_1	y_2	y_3	y_4	Кол-во наборов на $x_1 \dots x_8$
0	0	0	0	$1 \cdot 1 \cdot 1 \cdot 1 = 1$
0	0	0	1	$1 \cdot 1 \cdot 1 \cdot 3 = 3$
0	0	1	1	$1 \cdot 1 \cdot 3 \cdot 3 = 9$
0	1	1	1	$1 \cdot 3 \cdot 3 \cdot 3 = 27$
1	1	1	1	$3 \cdot 3 \cdot 3 \cdot 3 = 81$

Сложим количество наборов: $1 + 3 + 9 + 27 + 81 = 121$.

Ответ: 121.

Пример 7.25

Сколько существует различных наборов значений логических переменных $x_1, x_2, \dots, x_9, y_1, y_2, \dots, y_9$, которые удовлетворяют всем перечисленным ниже условиям?

$$(\neg (x_1 \equiv y_1)) \equiv (x_2 \equiv y_2)$$

$$(\neg (x_2 \equiv y_2)) \equiv (x_3 \equiv y_3)$$

...

$$(\neg (x_8 \equiv y_8)) \equiv (x_9 \equiv y_9).$$

В ответе не нужно перечислять все различные наборы значений переменных $x_1, x_2, \dots, x_9, y_1, y_2, \dots, y_9$, при которых выполнена данная система равенств. В качестве ответа вам нужно указать количество таких наборов.

Решение

Сделаем замену переменных:

$$(x_1 \equiv y_1) = z_1, (x_2 \equiv y_2) = z_2, \dots, (x_9 \equiv y_9) = z_9.$$

Систему можно записать в виде одного уравнения:

$$(\neg z_1 \equiv z_2) \wedge (\neg z_2 \equiv z_3) \wedge \dots \wedge (\neg z_8 \equiv z_9).$$

Эквивалентность истинна, только если оба операнда равны. Решениями этого уравнения будут два набора:

z_1	z_2	z_3	z_4	z_5	z_6	z_7	z_8	z_9
0	1	0	1	0	1	0	1	0
1	0	1	0	1	0	1	0	1

Так как $z_i = (x_i \equiv y_i)$, то значению $z_i = 0$ соответствуют два набора (x_i, y_i) : $(0,1)$ и $(1,0)$, а значению $z_i = 1$ — два набора (x_i, y_i) : $(0,0)$ и $(1,1)$.

Тогда первому набору $z_1, z_2, \dots, z_9$ соответствует 2^9 наборов $(x_1, y_1), (x_2, y_2), \dots, (x_9, y_9)$.

Столько же соответствует второму набору $z_1, z_2, \dots, z_9$. Тогда всего $2^9 + 2^9 = 1024$ наборов.

Ответ: 1024.

Решение систем логических уравнений методом визуального определения рекурсии

Этот метод применяется, если система уравнений достаточно проста и порядок увеличения количества наборов при добавлении переменных очевиден.

Пример 7.26

Сколько различных решений имеет система уравнений

$$\neg x_1 \vee x_2 = 1$$

$$\neg x_2 \vee x_3 = 1$$

...

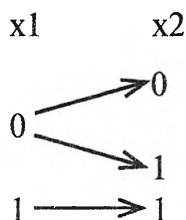
$$\neg x_9 \vee x_{10} = 1,$$

где $x_1, x_2, \dots, x_{10}$ — логические переменные?

В ответе не нужно перечислять все различные наборы значений $x_1, x_2, \dots, x_{10}$, при которых выполнена данная система равенств. В качестве ответа вам нужно указать количество таких наборов.

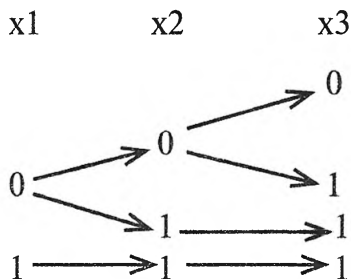
Решение

Решим первое уравнение. Дизъюнкция равна 1, если хотя бы один из ее операндов равен 1. То есть решениями являются наборы:



Для $x_1 = 0$ существуют два значения x_2 (0 и 1), а для $x_1 = 1$ только одно значение x_2 (1), такие, что набор (x_1, x_2) является решением уравнения. Всего 3 набора.

Добавим переменную x_3 и рассмотрим второе уравнение. Оно аналогично первому, значит для $x_2 = 0$ существуют два значения x_3 (0 и 1), а для $x_2 = 1$ только одно значение x_3 (1), такие, что набор (x_2, x_3) является решением уравнения. Всего 4 набора.



Несложно заметить, что при добавлении очередной переменной добавляется один набор. То есть рекурсивная формула количества наборов на $(i + 1)$ переменных:

$N_{i+1} = N_i + 1$. Тогда для десяти переменных получим 11 наборов.

Ответ: 11.

Решение систем логических уравнений различного типа

Пример 7.27

Сколько существует различных наборов значений логических переменных $x_1, \dots, x_4, y_1, \dots, y_4, z_1, \dots, z_4$, которые удовлетворяют всем перечисленным ниже условиям?

$$(x_1 \rightarrow x_2) \wedge (x_2 \rightarrow x_3) \wedge (x_3 \rightarrow x_4) = 1$$

$$(y_1 \rightarrow y_2) \wedge (y_2 \rightarrow y_3) \wedge (y_3 \rightarrow y_4) = 1$$

$$(z_1 \rightarrow z_2) \wedge (z_2 \rightarrow z_3) \wedge (z_3 \rightarrow z_4) = 1$$

$$x_4 \wedge y_4 \wedge z_4 = 0.$$

В ответе не нужно перечислять все различные наборы значений переменных $x_1, \dots, x_4, y_1, \dots, y_4, z_1, \dots, z_4$, при которых выполнена данная система равенств.

В качестве ответа вам нужно указать количество таких наборов.

Решение

Заметим, что три уравнения системы одинаковы на различных независимых наборах переменных.

Рассмотрим первое уравнение. Конъюнкция истинна (равна 1) только тогда, когда все ее операнды истинны (равны 1). Импликация равна 1 на всех наборах, кроме (1, 0). Значит, решением первого уравнения будут такие наборы x_1, x_2, x_3, x_4 , в которых 1 не стоит левее 0 (5 наборов):

x_1	x_2	x_3	x_4
0	0	0	0
0	0	0	1
0	0	1	1
0	1	1	1
1	1	1	1

Аналогично, решениями второго и третьего уравнений будут абсолютно такие же наборы $y_1, \dots, y_4$ и $z_1, \dots, z_4$.

Теперь проанализируем четвертое уравнение системы: $x_4 \wedge y_4 \wedge z_4 = 0$. Решением будут все наборы x_4, y_4, z_4 , в которых хотя бы одна из переменных равна 0.

То есть для $x_4 = 0$ подойдут все возможные наборы (y_4, z_4) , а для $x_4 = 1$ подойдут наборы (y_4, z_4) , в которых присутствует хотя бы один ноль: $(0, 0), (0, 1), (1, 0)$.

x_1	x_2	x_3	x_4	Кол-во наборов (y_4, z_4)
0	0	0	0	$5 \cdot 5 = 25$
0	0	0	1	$1 + 4 + 4 = 9$
0	0	1	1	$1 + 4 + 4 = 9$
0	1	1	1	$1 + 4 + 4 = 9$
1	1	1	1	$1 + 4 + 4 = 9$

Общее количество наборов $25 + 4 \cdot 9 = 25 + 36 = 61$.

Ответ: 61.

Пример 7.28

Сколько существует различных наборов значений логических переменных $x_1, x_2, x_3, x_4, y_1, y_2, y_3, y_4, z_1, z_2, z_3, z_4$, которые удовлетворяют всем перечисленным ниже условиям?

$$(x_1 \rightarrow x_2) \wedge (x_2 \rightarrow x_3) \wedge (x_3 \rightarrow x_4) = 1$$

$$(\neg x_1 \wedge y_1 \wedge z_1) \vee (x_1 \wedge \neg y_1 \wedge z_1) \vee (x_1 \wedge y_1 \wedge \neg z_1) = 1$$

$$(\neg x_2 \wedge y_2 \wedge z_2) \vee (x_2 \wedge \neg y_2 \wedge z_2) \vee (x_2 \wedge y_2 \wedge \neg z_2) = 1$$

$$(\neg x_3 \wedge y_3 \wedge z_3) \vee (x_3 \wedge \neg y_3 \wedge z_3) \vee (x_3 \wedge y_3 \wedge \neg z_3) = 1$$

$$(\neg x_4 \wedge y_4 \wedge z_4) \vee (x_4 \wedge \neg y_4 \wedge z_4) \vee (x_4 \wedge y_4 \wedge \neg z_4) = 1.$$

В ответе не нужно перечислять все различные наборы значений переменных

$x_1, x_2, x_3, x_4, y_1, y_2, y_3, y_4, z_1, z_2, z_3, z_4$, при которых выполнена данная система равенств. В качестве ответа вам нужно указать количество таких наборов.



Решение

Заметим, что второе, третье, четвертое и пятое уравнения системы одинаковы на различных независимых наборах переменных.

Рассмотрим второе уравнение. Несложно заметить, что его решением будут следующие наборы x_1, y_1, z_1 (3 набора):

x_1	y_1	z_1
0	1	1
1	0	1
1	1	0

Аналогично, для третьего, четвертого и пятого уравнения решениями будут абсолютно такие же наборы $x_2, y_2, z_2, \dots, x_4, y_4, z_4$.

Количество наборов (x_i, y_i, z_i) , в которых $x_i = 0$, равно 1. Количество наборов (x_i, y_i, z_i) , в которых $x_i = 1$, равно 2.

Решением первого уравнения будут такие наборы x_1, x_2, x_3, x_4 , в которых 1 не стоит левее 0.

x_1	x_2	x_3	x_4	Кол-во наборов $x_1, y_1, z_1, \dots, x_4, y_4, z_4$
0	0	0	0	$1 \cdot 1 \cdot 1 \cdot 1 = 1$
0	0	0	1	$1 \cdot 1 \cdot 1 \cdot 2 = 2$
0	0	1	1	$1 \cdot 1 \cdot 2 \cdot 2 = 4$
0	1	1	1	$1 \cdot 2 \cdot 2 \cdot 2 = 8$
1	1	1	1	$2 \cdot 2 \cdot 2 \cdot 2 = 16$

Всего наборов $1 + 2 + 4 + 8 + 16 = 31$.

Ответ: 31.

Решение систем логических уравнений методом построения рекуррентных формул

Метод построения рекуррентных формул применяется при решении сложных систем, в которых порядок увеличения количества наборов неочевиден, а построение дерева невозможно из-за объемов.

Пример 7.29

Сколько существует различных наборов значений логических переменных $x_1, x_2, \dots, x_7, y_1, y_2, \dots, y_7$, которые удовлетворяют всем перечисленным ниже условиям?

$$(x_1 \vee y_1) \wedge ((x_2 \wedge y_2) \rightarrow (x_1 \wedge y_1)) = 1$$

$$(x_2 \vee y_2) \wedge ((x_3 \wedge y_3) \rightarrow (x_2 \wedge y_2)) = 1$$

...

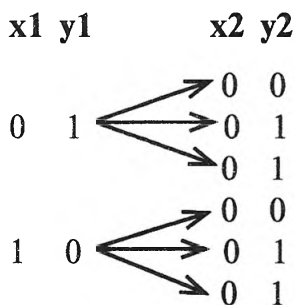
$$(x_6 \vee y_6) \wedge ((x_7 \wedge y_7) \rightarrow (x_6 \wedge y_6)) = 1$$

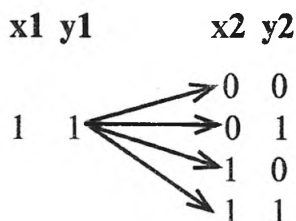
$$(x_7 \vee y_7) = 1.$$

В ответе не нужно перечислять все различные наборы значений переменных $x_1, x_2, \dots, x_7, y_1, y_2, \dots, y_7$, при которых выполнена данная система равенств. В качестве ответа вам нужно указать количество таких наборов.

Решение

Заметим, что первые шесть уравнений системы одинаковы и отличаются только набором переменных. Рассмотрим первое уравнение. Его решением будут следующие наборы переменных:





Обозначим:

число наборов (0,0) на переменных (x1, y1) через A_1 ,

число наборов (0,1) на переменных (x1, y1) через B_1 ,

число наборов (1,0) на переменных (x1, y1) через C_1 ,

число наборов (1,1) на переменных (x1, y1) через D_1 .

число наборов (0,0) на переменных (x2, y2) через A_2 ,

число наборов (0,1) на переменных (x2, y2) через B_2 ,

число наборов (1,0) на переменных (x2, y2) через C_2 ,

число наборов (1,1) на переменных (x2, y2) через D_2 .

Из дерева решений видим, что

$$A_1 = 0, B_1 = 1, C_1 = 1, D_1 = 1.$$

Заметим, что набор (0,0) на переменных (x2, y2) получается из наборов (0,1), (1,0) и (1,1) на переменных (x1, y1). То есть $A_2 = B_1 + C_1 + D_1$.

Набор (0,1) на переменных (x2, y2) получается из наборов (0,1), (1,0) и (1,1) на переменных (x1, y1).

То есть $B_2 = B_1 + C_1 + D_1$.

Аналогично рассуждая, заметим, что $C_2 = B_1 + C_1 + D_1$.
 $D_2 = D_1$.

Таким образом, получаем рекуррентные формулы:

$$A_{i+1} = B_i + C_i + D_i$$

$$B_{i+1} = B_i + C_i + D_i$$

$$C_{i+1} = B_i + C_i + D_i$$

$$D_{i+1} = A_i + B_i + C_i + D_i$$

Составим таблицу

Наборы	Обозн.	Формула	Количество наборов						
			i=1	i=2	i=3	i=4	i=5	i=6	i=7
(0,0)	A_i	$A_{i+1} = B_i + C_i + D_i$	0	3	7	15	31	63	127
(0,1)	B_i	$B_{i+1} = B_i + C_i + D_i$	1	3	7	15	31	63	127
(1,0)	C_i	$C_{i+1} = B_i + C_i + D_i$	1	3	7	15	31	63	127
(1,1)	D_i	$D_{i+1} = D_i$	1	1	1	1	1	1	1

Последнему уравнению $(x_7 \vee y_7) = 1$ удовлетворяют все наборы, кроме тех, в которых $x_7 = 0$ и $y_7 = 0$. В нашей таблице число таких наборов A_7 .

Тогда общее количество наборов равно $B_7 + C_7 + D_7 = 127 + 127 + 1 = 255$.

Ответ: 255.

Задача № 11

Рекурсивные алгоритмы

Рекурсия — это способ определения объектов (понятий), при котором определение объекта строится, опираясь на само понятие объекта.

Для того чтобы задать рекурсию, необходимо описать:

- условие остановки рекурсии (базовый случай);
- рекуррентную формулу.

В программировании, если процедура вызывает сама себя, то, по сути, это приводит к повторному выполнению содержащихся в ней инструкций, что аналогично работе цикла. Рекурсия позволяет заменить цикл и в некоторых сложных задачах делает решение более понятным, хотя часто менее эффективным.

Некоторые языки программирования не содержат циклических конструкций вовсе, предоставляя программи-

стам организовывать повторения с помощью рекурсии (например, Пролог, где рекурсия — основной прием программирования).

Классическим примером рекурсивного алгоритма является описание вычисления факториала:

$$F(n) := \begin{cases} 1, & n = 0 \\ n * F(n-1), & n > 0, \end{cases}$$

где $F(n-1) = (n-1)!$

Рекурсивные алгоритмы вычисления одной функции

Пример 7.30

Алгоритм вычисления значения функции $F(n)$, где n — натуральное число, задан следующими рекуррентными соотношениями:

$$F(n) = 1 \text{ при } n = 1;$$

$$F(n) = F(n-1) \cdot n \text{ при } n \geq 2.$$

Чему равно значение функции $F(6)$?

В ответе запишите только натуральное число.

Решение

Последовательно найдем значения функции от базового случая $F(1)$ до искомого значения $F(6)$:

$$F(1) = 1;$$

$$F(2) = 2;$$

$$F(3) = 6;$$

$$F(4) = 24;$$

$$F(5) = 120;$$

$$F(6) = 720.$$

Ответ: 720.

Пример 7.31

Алгоритм вычисления значения функции $F(n)$, где n — натуральное число, задан следующими соотношениями:

$$F(1) = 1;$$

$$F(n) = F(n-1) + 2^{n-1}, \text{ если } n > 1.$$

Чему равно значение функции $F(12)$?

В ответе запишите только натуральное число.

Решение

Последовательно найдем значения функции от базового случая $F(1)$ до искомого значения $F(12)$:

$$F(1) = 1;$$

$$F(2) = 3;$$

$$F(3) = 7;$$

$$F(4) = 15;$$

$$F(5) = 31;$$

$$F(6) = 63;$$

$$F(7) = 127;$$

$$F(8) = 255;$$

$$F(9) = 511;$$

$$F(10) = 1023;$$

$$F(11) = 2047;$$

$$F(12) = 4095.$$

Ответ: 4095.

Пример 7.32

Алгоритм вычисления значения функции $F(n)$, где n — натуральное число, задан следующими соотношениями:

$$F(1) = 3; F(2) = 3;$$

$$F(n) = 5 \cdot F(n-1) - 4 \cdot F(n-2) \text{ при } n > 2.$$

Чему равно значение функции $F(15)$? В ответе запишите только натуральное число.

Решение

Последовательно найдём значения функции от базовых случаев $F(1)$ и $F(2)$ до искомого значения $F(15)$:

$$F(1) = 3;$$

$$F(2) = 3;$$

$$F(3) = 15 - 12 = 3;$$

$$F(4) = 15 - 12 = 3;$$

$$F(5) = 15 - 12 = 3;$$

...

$$F(15) = 3.$$

Ответ: 3.

Рекурсивные алгоритмы вычисления нескольких функций

Пример 7.33

Алгоритм вычисления значений функций $F(n)$ и $G(n)$, где n — натуральное число, задан следующими соотношениями:

$$F(1) = 1; G(1) = 1;$$

$$F(n) = F(n-1) - G(n-1);$$

$$G(n) = F(n-1) + 2 \cdot G(n-1), \text{ при } n \geq 2.$$

Чему равно значение величины $F(5)/G(5)$? В ответе запишите только целое число.

Решение

Последовательно найдём значения функций от базового случая $F(1)$, $G(1)$ до искомым значений $F(5)$, $G(5)$:

$$F(1) = 1; G(1) = 1;$$

$$F(2) = F(1) - G(1) = 1 - 1 = 0;$$

$$G(2) = F(1) + 2 \cdot G(1) = 1 + 2 = 3;$$

$$F(3) = F(2) - G(2) = 0 - 3 = -3;$$

$$G(3) = F(2) + 2 \cdot G(2) = 0 + 6 = 6;$$

$$F(4) = F(3) - G(3) = -3 - 6 = -9;$$

$$G(4) = F(3) + 2 \cdot G(3) = -3 + 12 = 9;$$

$$F(5) = F(4) - G(4) = -9 - 9 = -18;$$

$$G(5) = F(4) + 2 \cdot G(4) = -9 + 18 = 9.$$

$$F(5)/G(5) = -18/9 = -2.$$

Ответ: -2 .

Рекурсивные алгоритмы выполнения процедур

Пример 7.34

Ниже на пяти языках программирования записан рекурсивный алгоритм F .

Бейсик	Python
<pre>SUB F(n) PRINT n IF n < 5 THEN F(n + 1) F(n + 3) ENDIF END SUB</pre>	<pre>def F(n): print(n) if n < 5: F(n + 1) F(n + 3)</pre>
Паскаль	Алгоритмический язык
<pre>procedure F(n: integer); begin writeln(n); if n < 5 then begin F(n + 1); F(n + 3) end end</pre>	<pre>алг F(цел n) нач вывод n если n < 5 то F(n + 1) F(n + 3) все кон</pre>
Си	
<pre>void F(int n) { printf("%d\n", n); if (n < 5) { F(n + 1); F(n + 3); } }</pre>	

Чему равна сумма всех чисел, напечатанных на экране, при выполнении вызова $F(1)$?

Решение

Выпишем последовательно все действия, которые выполняют запускаемые процедуры:

$F(1)$ выполнит следующие действия:

Вывод числа 1, $F(2)$, $F(4)$.

$F(2)$ выполнит следующие действия:

Вывод числа 2, $F(3)$, $F(5)$.

$F(4)$ выполнит следующие действия:

Вывод числа 4, $F(5)$, $F(7)$.

$F(3)$ выполнит следующие действия:

Вывод числа 3, $F(4)$, $F(6)$.

$F(5)$ выполнит следующие действия: Вывод числа 5.

$F(5)$ выполнит следующие действия: Вывод числа 5.

$F(7)$ выполнит следующие действия: Вывод числа 7.

$F(4)$ выполнит следующие действия:

Вывод числа 4, $F(5)$, $F(7)$.

$F(6)$ выполнит следующие действия: Вывод числа 6.

$F(5)$ выполнит следующие действия: Вывод числа 5.

$F(7)$ выполнит следующие действия: Вывод числа 7.

Просуммируем все числа, выведенные на экран:

$$1 + 2 + 4 + 3 + 5 + 5 + 7 + 4 + 6 + 5 + 7 = 49.$$

Ответ: 49.

Пример 7.35

Ниже на пяти языках программирования записаны две рекурсивные функции (процедуры): F и G .

Бейсик	Python
<pre> DECLARE SUB F(n) DECLARE SUB G(n) SUB F(n) IF n > 0 THEN G(n-1) END SUB SUB G(n) PRINT " * " IF n > 1 THEN F(n-3) END SUB </pre>	<pre> def F(n): if n > 0: G(n-1) def G(n): print(" * ") if n > 1: F(n-3) </pre>
Алгоритмический язык	Паскаль
<pre> алг F(цел n) нач если n > 0 то G(n-1) все кон алг G(цел n) нач вывод " * " если n > 1 то F(n-3) все кон </pre>	<pre> procedure F(n: integer); forward; procedure G(n: integer); forward; procedure F(n: integer); begin if n > 0 then G(n-1); end; procedure G(n: integer); begin writeln(" * "); if n > 1 then F(n-3); end </pre>
Си	
<pre> void F(int n); void G(int n); void F(int n) { if (n > 0) G(n-1); } void G(int n) { printf(" * "); if (n > 1) F(n-3); } </pre>	

Сколько символов «звёздочка» будет напечатано на экране при выполнении вызова $F(11)$?

Решение

Выпишем последовательно все действия, которые выполнят запускаемые процедуры:

$F(11)G(10) * F(7)G(6) * F(3)G(2) * F(-1)$

Всего на экране будет напечатано 3 «звездочки».

Ответ: 3.

Пример 7.36

Дан рекурсивный алгоритм:

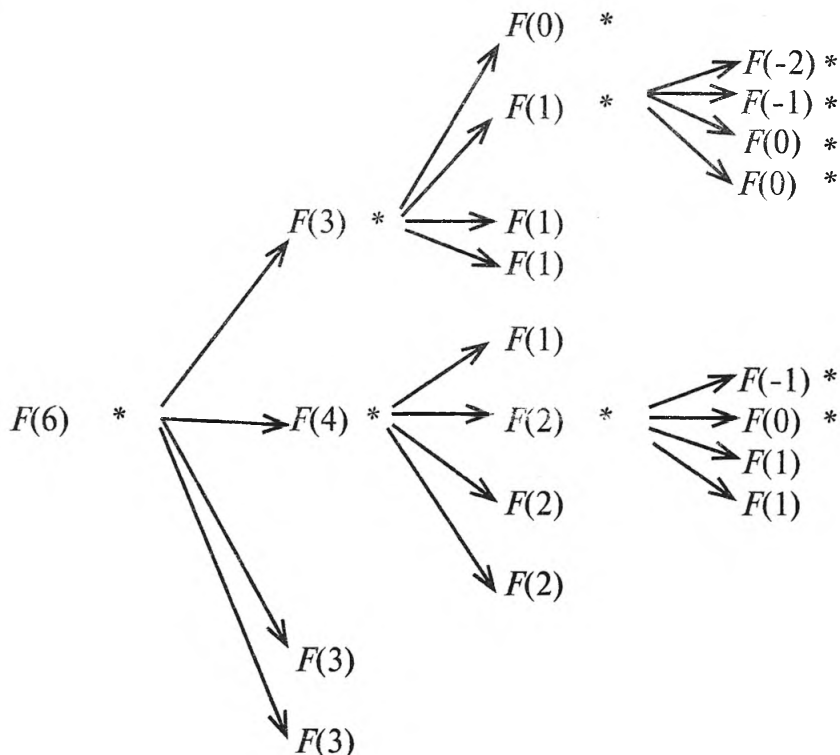
```
procedure F(n: integer);  
begin  
  writeln('*');  
  if n > 0 then begin  
    F(n-3);  
    F(n-2);  
    F(n div 2);  
    F(n div 2);  
  end  
end;
```

Сколько символов «звездочка» будет напечатано на экране при выполнении вызова $F(6)$?

Решение

Для наглядности изобразим схему работы алгоритма в виде дерева.

Причем распишем до конца каждое значение $F(n)$ только один раз. Например, расписав один раз $F(1)$, мы видим, что она напечатает в результате 5 звездочек. То есть $F(1) = 5$.



Проанализировав дерево, видим, что

$$F(0) = 1;$$

$$F(2) = 3 + 2 * F(1) = 13;$$

$$F(3) = 1 + F(0) + 3 * F(1) = 1 + 1 + 15 = 17;$$

$$F(4) = 1 + F(1) + 3 * F(2) = 1 + 5 + 3 * 13 = 45;$$

$$F(6) = 1 + 3 * F(3) + F(4) = 1 + 3 * 17 + 45 = 46 + 51 = 97.$$

Ответ: 97.

Задача № 19

Работа с массивами

Массив — это тип или структура данных в виде набора компонентов (элементов массива), расположенных в памяти непосредственно друг за другом. При этом доступ к

отдельным элементам массива осуществляется с помощью индексации, т.е. ссылки на массив с указанием номера (индекса) нужного элемента. Размерность массива — это количество индексов, необходимое для однозначного доступа к элементу массива.

Одномерный массив представляет собой пронумерованную последовательность элементов одного и того же типа, имеющих общее имя. Для обращения к элементу массива используют имя и порядковый номер элемента в квадратных скобках: $A[i]$.

Двумерный массив представляет собой матрицу элементов одного и того же типа, в которой элемент, расположенный на пересечении i -й строки и j -го столбца, обозначается $A[i, j]$.

Одномерные массивы

Пример 7.37

В программе описан одномерный целочисленный массив A с индексами от 0 до 10. Ниже представлен фрагмент этой программы, записанный на разных языках программирования, в котором значения элементов массива сначала задаются, а затем меняются.

Бейсик	Паскаль
FOR i=0 TO 10 $A(i)=i-1$ NEXT i FOR i=1 TO 10 $A(i-1)=A(i)$ NEXT i $A(10)=10$	for i:=0 to 10 do $A[i]:=i-1$; for i:=1 to 10 do $A[i-1]:=A[i]$; $A[10]:=10$;

Си	Алгоритмический язык
<pre> for (i = 0; i <= 10; i++) A[i]=i-1; for (i = 1; i <= 10; i++) A[i-1]=A[i]; A[10]=10; </pre>	<pre> нц для i от 0 до 10 A[i]:=i-1 кц нц для i от 1 до 10 A[i-1]:=A[i] кц A[10]:=10 </pre>

Как изменятся элементы этого массива после выполнения фрагмента программы?

- 1) все элементы, кроме последнего, окажутся равны между собой;
- 2) все элементы окажутся равны своим индексам;
- 3) все элементы, кроме последнего, будут сдвинуты на один элемент вправо;
- 4) все элементы, кроме последнего, уменьшатся на единицу.

Решение

Выполним последовательно все действия, описанные в программе. После первого цикла массив примет вид:

Индекс	0	1	2	3	4	5	6	7	8	9	10
Значение	-1	0	1	2	3	4	5	6	7	8	9

Второй цикл сдвигает элементы массива на один влево:

$i = 1, A[0] := A[1] = 0$

$i = 2, A[1] := A[2] = 1$

...

$i = 10, A[9] := A[10] = 9$

А затем десятому элементу присваивается значение 10:
 $A[10] := 10$:

Индекс	0	1	2	3	4	5	6	7	8	9	10
Значение	0	1	2	3	4	5	6	7	8	9	10

Правильный ответ указан под номером 2.

Ответ: 2.

Пример 7.38

В программе используется одномерный целочисленный массив A с индексами от 0 до 9. Значения элементов равны 4, 7, 3, 8, 5, 0, 1, 2, 9, 6 соответственно, т.е. $A[0] = 4$, $A[1] = 7$ и т.д.

Определите значение переменной C после выполнения следующего фрагмента этой программы (записанного ниже на пяти языках программирования).

Бейсик	Python
<pre> C = 0 FOR i=0 TO 9 IF A(i) < A(0) THEN C = C + 1 t = A(i) A(i) = A(0) A(0) = t END IF NEXT i </pre>	<pre> C = 0 for i in range(1,10) if A[i] < A[0]: C = C + 1 t = A[i] A[i] = A[0] A[0] = t </pre>
Алгоритмический язык	Паскаль
<pre> C := 0 нц для i от 1 до 9 если A[i] < A[0] то C := C + 1 t := A[i] A[i] := A[0] A[0] := t все кон </pre>	<pre> C := 0; for i := 1 to 9 do if A[i] < A[0] then begin C := C + 1; t := A[i]; A[i] := A[0]; A[0] := t; end; end; </pre>
Си	
<pre> C := 0; for (i = 1; i < 10; i++) if (A[i] < A[0]) { C++; t = A[i]; A[i] = A[0]; A[0] = t; } </pre>	

Решение

Если i -й элемент массива меньше нулевого, то программа меняет их местами и увеличивает значение переменной s на 1.

Индекс	0	1	2	3	4	5	6	7	8	9
Значение	4	7	3	8	5	0	1	2	9	6

Первый раз условие « i -й элемент массива меньше нулевого» выполнится при $i = 2$. Массив примет вид:

Индекс	0	1	2	3	4	5	6	7	8	9
Значение	3	7	4	8	5	0	1	2	9	6

А переменная s примет значение 1.

Второй раз условие « i -й элемент массива меньше нулевого» выполнится при $i = 5$. Массив примет вид:

Индекс	0	1	2	3	4	5	6	7	8	9
Значение	0	7	4	8	5	3	1	2	9	6

А переменная s примет значение 2.

Больше условие « i -й элемент массива меньше нулевого» не выполнится ни разу. Значит, $s = 2$.

Ответ: 2.

Пример 7.39

В программе описаны одномерный целочисленный массив A с индексами от 0 до 10 и целочисленные переменные i и t . Ниже представлен фрагмент этой программы, записанный на разных языках программирования.

Бейсик	Паскаль
FOR i = 0 TO 10 A(i) = i NEXT i t = A(10) FOR i = 9 TO 0 STEP -1 A(i+1) = A(i) NEXT i A(0) = t	for i := 0 to 10 do A[i] := i; t := A[10] for i := 9 down to 0 do A[i+1] := A[i]; A[0] := t;
Си	Алгоритмический язык
for (i = 0; i <= 10; i++) A[i] = i; t = A[10]; for (i = 9; i >= 0; i--) A[i+1] = A[i]; A[0] = t;	нц для i от 0 до 10 A[i] := i кц t := A[10] нц для i от 9 до 0 шаг -1 A[i+1] := A[i] кц A[0] := t

Чему окажутся равны элементы этого массива после выполнения фрагмента программы?

- 1) 10 0 0 0 0 0 0 0 0 0 0
- 2) 10 2 3 4 5 6 7 8 9 10 10
- 3) 10 10 10 10 10 10 10 10 10 10 10
- 4) 10 0 1 2 3 4 5 6 7 8 9

Решение

Выполним последовательно все действия, описанные в программе. После первого цикла массив примет вид:

Индекс	0	1	2	3	4	5	6	7	8	9	10
Значение	0	1	2	3	4	5	6	7	8	9	10

Второй цикл работает в обратном порядке, от 9 до 0:

i = 9: A[10] := A[9] = 9

i = 8: A[9] := A[8] = 8

...

$i = 1: A[2] := A[1] = 1$

$i = 0: A[1] := A[0] = 0$

А затем нулевому элементу присваивается значение 10 ($A[0] := 10$):

Индекс	0	1	2	3	4	5	6	7	8	9	10
Значение	10	0	1	2	3	4	5	6	7	8	9

Правильный ответ указан под номером 4.

Ответ: 4.

Пример 7.40

В программе описан одномерный целочисленный массив с индексами от 0 до 10. В приведенном ниже фрагменте программы массив сначала заполняется, а потом изменяется:

```
for i:=0 to 10 do A[i]:=i;
```

```
  for i:=0 to 10 do begin
```

```
    A[10-i]:=A[i];
```

```
    A[i]:=A[10-i];
```

```
  end;
```

Чему будут равны элементы этого массива?

1) 10 9 8 7 6 5 4 3 2 1 0

2) 0 1 2 3 4 5 6 7 8 9 10

3) 10 9 8 7 6 5 6 7 8 9 10

4) 0 1 2 3 4 5 4 3 2 1 0

Решение

Выполним последовательно все действия, описанные в программе. После первого цикла элементы массива равны номерам индексов:

Индекс	0	1	2	3	4	5	6	7	8	9	10
Значение	0	1	2	3	4	5	6	7	8	9	10

Второй цикл меняет значение элементов массива следующим образом:

$i = 0, A[10] := A[0] = 0, A[0] := A[10] = 0$

$i = 1, A[9] := A[1] = 1, A[1] := A[9] = 1$

$i = 2, A[8] := A[2] = 2, A[2] := A[8] = 2$

$i = 3, A[7] := A[3] = 3, A[3] := A[7] = 3$

$i = 4, A[6] := A[4] = 4, A[4] := A[6] = 4$

$i = 5, A[5] := A[5] = 5, A[5] := A[5] = 5$

$i = 6, A[6] := A[4] = 4, A[4] := A[6] = 4$

$i = 7, A[7] := A[3] = 3, A[3] := A[7] = 3$

$i = 8, A[8] := A[2] = 2, A[2] := A[8] = 2$

$i = 9, A[9] := A[1] = 1, A[1] := A[9] = 1$

$i = 10, A[10] := A[0] = 0, A[0] := A[10] = 0$

Индекс	0	1	2	3	4	5	6	7	8	9	10
Значение	0	1	2	3	4	5	4	7	2	1	0

Ответ: 4.

Пример 7.41

Значения двух массивов $A[1..100]$ и $B[1..100]$ задаются с помощью следующего фрагмента программы:

for $n:=1$ to 100 do

$A[n] := n - 10;$

for $n:=1$ to 100 do

$B[n] := A[n]*n;$

Сколько элементов массива B будут иметь положительные значения?

Решение

Первый цикл заполняет массив А:

$$n = 1, A[1] := 1 - 10 = -9$$

$$n = 2, A[2] := 2 - 10 = -8$$

$$n = 3, A[3] := 3 - 10 = -7$$

...

$$n = 10, A[10] := 10 - 10 = 0$$

...

$$n = 100, A[100] := 100 - 10 = 90$$

То есть 9 первых элементов массива А отрицательные, 10-й элемент равен 0, остальные 90 элементов положительные.

Второй цикл заполняет массив В:

$$n = 1, B[1] := A[1] * 1 = -9 * 1 = -9$$

$$n = 2, B[2] := A[2] * 2 = -8 * 2 = -16$$

...

$$n = 10, B[10] := A[10] * 10 = 0 * 10 = 0$$

...

$$n = 100, B[100] := A[100] * 100 = 90 * 100 = 9000$$

То есть положительным элементам массива А соответствуют положительные элементы массива В. Всего их 90.

Ответ: 90.

Двумерные массивы**Пример 7.42**

Значения элементов двумерного массива А размером 5х5 задаются с помощью вложенного цикла в представленном фрагменте программы:

for i:=1 to 5 do

for j:=1 to 5 do begin

A[i,j] := i*j;

end;

Сколько элементов массива будут иметь значения больше 10?

Решение

Выполним последовательно все действия, описанные в программе:

$i = 1$ (Заполняем первую строку матрицы)

$$j = 1, A[1,1] := 1 * 1 = 1$$

$$j = 2, A[1,2] := 1 * 2 = 2$$

$$j = 3, A[1,3] := 1 * 3 = 3$$

$$j = 4, A[1,4] := 1 * 4 = 4$$

$$j = 5, A[1,5] := 1 * 5 = 5$$

	1	2	3	4	5
1	1	2	3	4	5
2					
3					
4					
5					

$i = 2$ (Заполняем вторую строку матрицы)

$$j = 1, A[2,1] := 2 * 1 = 2$$

$$j = 2, A[2,2] := 2 * 2 = 4$$

$$j = 3, A[2,3] := 2 * 3 = 6$$

$$j = 4, A[2,4] := 2 * 4 = 8$$

$$j = 5, A[2,5] := 2 * 5 = 10$$

	1	2	3	4	5
1	1	2	3	4	5
2	2	4	5	8	10
3					
4					
5					

.....

$i = 5$ (Заполняем пятую строку матрицы)

$j = 1, A[5,1] := 5 * 1 = 5$

$j = 2, A[5,2] := 5 * 2 = 10$

$j = 3, A[5,3] := 5 * 3 = 15$

$j = 4, A[5,4] := 5 * 4 = 20$

$j = 5, A[5,5] := 5 * 5 = 25$

Значение элементов матрицы равно произведению номеров его строки и столбца

	1	2	3	4	5
1	1	2	3	4	5
2	2	4	5	8	10
3	3	6	9	12	15
4	4	8	12	16	20
5	5	10	15	20	25

Значение больше 10 имеют 8 элементов матрицы.

Ответ: 8.

Пример 7.43

Значения элементов двумерного массива

$A[1..100,1..100]$

задаются с помощью следующего фрагмента программы:

```
for i:=1 to 100 do
```

```
  for k:=1 to 100 do
```

```
    if i > k then
```

```
      A[i,k] := 1
```

```
    else A[i,k] := -1;
```

Чему равна сумма элементов массива после выполнения этого фрагмента программы?

Решение

В данной задаче нет необходимости выписывать значения каждого элемента матрицы. Достаточно заметить, что элементу присваивается значение 1, если номер его строки больше номера столбца, т.е. если элемент расположен ниже главной диагонали матрицы. Элементам главной диагонали и элементам, расположенным выше, присваивается значение -1 .

	1	2	3	...			100
1	-1	-1	-1				-1
2	1	-1	-1				-1
3	1	1	-1				-1
	1	1	1	-1			-1
...					...		-1
						-1	...
100	1	1	1	...	1	1	-1

Тогда элементы, расположенные выше и ниже главной диагонали, при суммировании дадут ноль, а сумма элементов, расположенных на главной диагонали, равна -100 .

Ответ: -100 .

Задача № 26

Построение дерева игры.

Поиск выигрышной стратегии

В простых играх можно найти выигрышную стратегию, расписав все возможные ходы игроков. Такая схема ходов называется деревом игры.

Все позиции в простых играх делятся на выигрышные и проигрышные.

Выигрышная позиция — это такая позиция, в которой игрок, делающий первый ход, может гарантированно выиграть при любой игре соперника. При этом алгоритм выбора очередного хода, приводящего к выигрышу, называется выигрышной стратегией. Считается, что игрок, обладающий выигрышной стратегией, не ошибается.

Проигрышная позиция — это такая позиция, при которой игрок, делающий первый ход, проигрывает независимо от выбора очередного хода.

Определение выигравшего игрока при заданной начальной позиции

Пример 7.44

Два игрока играют в следующую игру. Перед ними лежат две кучи камней, в первой из которых 3, а во второй — 6 камней. У каждого игрока неограниченно много камней. Игроки ходят по очереди. Ход состоит в том, что игрок или удваивает число камней в какой-то куче, или добавляет 2 камня в какую-то кучу. Выигрывает игрок, после хода которого общее число камней в двух кучах становится не менее 24 камней. Кто выигрывает при безошибочной игре обоих игроков — игрок, делающий первый ход, или игрок, делающий второй ход? Каким должен быть первый ход выигравшего игрока? Ответ обоснуйте.

Решение

Для доказательства выигрыша нам достаточно привести неполное дерево игры, в котором рассмотрены все возможные ходы проигравшего игрока и одна любая, приводящая к выигрышу, последовательность ходов выигравшего игрока.

В приведенной таблице числа, разделенные запятой, соответствуют количеству камней в первой и второй кучах соответственно.



	2 ход	3 ход	4 ход	5 ход	
Позиция после первого хода	II игрок (все варианты хода)	I игрок (выигрышный ход)	II игрок (все варианты хода)	I игрок (один из вариантов)	Пояснение
5,6	5,8	7,8	14,8	28,8	Первый игрок выигрывает на пятом ходу, после любого ответа второго игрока, например, удвоив число камней в самой большой куче
			9,8	18,8	
			7,16	7,32	
			7,10	7,20	
	7,6	7,8	Те же варианты четвертого и пятого ходов		
	5,12	5,24	Первый игрок выиграл		
	10,6	20,6	Первый игрок выиграл		

Выигрывает первый игрок. Своим первым ходом он должен добавить 2 камня в первую кучу.

Таблица содержит все возможные варианты ходов второго игрока и ходы, приводящие к победе первого.

Ответ: Выигрывает первый игрок. Своим первым ходом он должен добавить 2 камня в первую кучу.

Пример 7.45

Два игрока, Петя и Ваня, играют в следующую игру. Перед ними лежат две кучки камней, в первой из которых 4, а во второй — 3 камня. У каждого игрока неограниченно много камней. Игроки ходят по очереди, первый ход делает Петя. Ход состоит в том, что игрок или утраивает число камней в какой-то куче, или добавляет 1 камень в какую-то кучу. Игра завершается в тот момент, когда общее количество камней в двух кучах становится не менее 20. Если в момент завершения игры общее число камней в двух кучах не менее 35, то

выиграл Ваня, в противном случае — Петя. Кто выигрывает при безошибочной игре обоих игроков? Укажите стратегию выигрывающего игрока — какой ход он должен сделать в каждой из позиций, которые могут ему встретиться при правильной игре. Докажите, что описанная стратегия — выигрышная.

Решение

Выигрывает Петя. Для того чтобы выиграть, он должен утроить количество камней во второй куче.

Для доказательства рассмотрим неполное дерево игры, оформленное в виде таблицы, где в каждой ячейке записаны пары чисел, разделённые запятой. Эти числа соответствуют количеству камней на каждом этапе игры в первой и второй кучах соответственно.

1 ход	2 ход	3 ход	
Петя (выигрышный ход)	Ваня (все варианты)	Петя (выигрышный ход)	
	12,9	Петя выиграл	
	4,27	Петя выиграл	
	5,9	15,9	Петя выиграл
		5,27	
	4,10	12,10	Петя выиграл
		4,30	

Таблица содержит все возможные варианты ходов второго игрока. Из неё видно, что при любом ходе второго игрока у первого имеется ход, приводящий к победе. Причём у первого игрока есть два варианта выигрышного хода. Описание любого из них является правильным решением.

Ответ: Выигрывает Петя. Для того чтобы выиграть, он должен утроить количество камней во второй куче.

Определение выигравшего игрока для различных начальных позиций

Пример 7.46

Два игрока, Петя и Ваня, играют в следующую игру. Перед игроками лежат две кучи камней. Игроки ходят по очереди, первый ход делает Петя. За один ход игрок может добавить в одну из куч (по своему выбору) один камень или увеличить количество камней в куче в два раза. Например, пусть в одной куче 10 камней, а в другой 7 камней; такую позицию в игре будем обозначать $(10, 7)$. Тогда за один ход можно получить любую из четырёх позиций: $(11, 7)$, $(20, 7)$, $(10, 8)$, $(10, 14)$. Для того чтобы делать ходы, у каждого игрока есть неограниченное количество камней.

Игра завершается в тот момент, когда суммарное количество камней в кучах становится не менее 73. Победителем считается игрок, сделавший последний ход, т.е. первым получивший такую позицию, что в кучах всего будет 73 камня или больше.

Будем говорить, что игрок имеет выигрышную стратегию, если он может выиграть при любых ходах противника. Описать стратегию игрока — значит описать, какой ход он должен сделать в любой ситуации, которая ему может встретиться при различной игре противника. Например, при начальных позициях $(6, 34)$, $(7, 33)$, $(9, 32)$ выигрышная стратегия есть у Пети. Чтобы выиграть, ему достаточно удвоить количество камней во второй куче.

Задание 1. Для каждой из начальных позиций $(6, 33)$, $(8, 32)$ укажите, кто из игроков имеет выигрышную стратегию. В каждом случае опишите выигрышную стратегию; объясните, почему эта стратегия ведёт к выигрышу, и укажите, какое наибольшее количество ходов может потребоваться победителю для выигрыша при этой стратегии.

Задание 2. Для каждой из начальных позиций (6, 32), (7, 32), (8, 31) укажите, кто из игроков имеет выигрышную стратегию. В каждом случае опишите выигрышную стратегию; объясните, почему эта стратегия ведёт к выигрышу, и укажите, какое наибольшее количество ходов может потребоваться победителю для выигрыша при этой стратегии.

Задание 3. Для начальной позиции (7, 31) укажите, кто из игроков имеет выигрышную стратегию. Опишите выигрышную стратегию; объясните, почему эта стратегия ведёт к выигрышу, и укажите, какое наибольшее количество ходов может потребоваться победителю для выигрыша при этой стратегии. Постройте дерево всех партий, возможных при указанной Вами выигрышной стратегии. Представьте дерево в виде рисунка или таблицы.

Решение

Задание 1. В начальных позициях (6, 33), (8, 32) выигрышная стратегия есть у Вани. При начальной позиции (6, 33) после первого хода Пети может получиться одна из следующих четырёх позиций: (7, 33), (12, 33), (6, 34), (6, 66). Каждая из этих позиций содержит менее 73 камней. При этом из любой из этих позиций Ваня может получить позицию, содержащую не менее 73 камней, удвоив количество камней во второй куче. Для позиции (8, 32) после первого хода Пети может получиться одна из следующих четырёх позиций: (9, 32), (16, 32), (8, 33), (8, 64). Каждая из этих позиций содержит менее 73 камней. При этом из любой из этих позиций Ваня может получить позицию, содержащую не менее 73 камней, удвоив количество камней во второй куче. Таким образом, Ваня при любом ходе Пети выигрывает своим первым ходом.

Задание 2. В начальных позициях (6, 32), (7, 32) и (8, 31) выигрышная стратегия есть у Пети. При начальной позиции (6, 32) он должен первым ходом получить позицию (6, 33), из начальных позиций (7, 32) и (8, 31) Петя после первого

хода должен получить позицию (8, 32). Позиции (6, 33) и (8, 32) рассмотрены при разборе задания 1. В этих позициях выигрышная стратегия есть у игрока, который будет ходить вторым (теперь это Петя). Эта стратегия описана при разборе задания 1. Таким образом, Петя при любой игре Вани выигрывает своим вторым ходом.

Задание 3. В начальной позиции (7, 31) выигрышная стратегия есть у Вани. После первого хода Пети может возникнуть одна из четырёх позиций: (8, 31), (7, 32), (14, 31) и (7, 62). В позициях (14, 31) и (7, 62) Ваня может выиграть одним ходом, удвоив количество камней во второй куче. Позиции (8, 31) и (7, 32) были рассмотрены при разборе задания 2. В этих позициях у игрока, который должен сделать ход (теперь это Ваня), есть выигрышная стратегия. Эта стратегия описана при разборе задания 2. Таким образом, в зависимости от игры Пети Ваня выигрывает на первом или втором ходу.

Исходное положение	Положения после очередных ходов			
	1-й ход Пети (разобраны все ходы, указана полученная позиция)	1-й ход Вани (только ход по стратегии, указана полученная позиция)	2-й ход Пети (разобраны все ходы, указана полученная позиция)	1-й ход Вани (только ход по стратегии, указана полученная позиция)
(7, 31) Всего: 38	(7, 31 + 1) = (7, 32) Всего: 39	(7 + 1, 32) = (8, 32) Всего: 40	(8 + 1, 32) = (9, 32) Всего: 41	(9, 32 · 2) = (9, 64) Всего: 73
			(8, 32 + 1) = (8, 33) Всего: 41	(9, 33 · 2) = (8, 66) Всего: 74
			(8 · 2, 32) = (16, 32) Всего: 48	(16, 32 · 2) = (16, 64) Всего: 80
			(8, 32 · 2) = (8, 64) Всего: 72	(8, 64 · 2) = (8, 128) Всего: 136

Исходное положение	Положения после очередных ходов			
	1-й ход Пети (разобраны все ходы, указана полученная позиция)	1-й ход Вани (только ход по стратегии, указана полученная позиция)	2-й ход Пети (разобраны все ходы, указана полученная позиция)	1-й ход Вани (только ход по стратегии, указана полученная позиция)
	$(7 + 1, 31) =$ $= (8, 31)$ Всего: 39	$(8, 31 + 1) =$ $= (8, 32)$ Всего: 40	$(8 + 1, 32) =$ $(9, 32)$ Всего: 41	$(9, 32 \cdot 2) =$ $(9, 64)$ Всего: 73
			$(8, 32 + 1) =$ $(8, 33)$ Всего: 41	$(9, 33 \cdot 2) =$ $(8, 66)$ Всего: 74
			$(8 \cdot 2, 32) =$ $(16, 32)$ Всего: 48	$(16, 32 \cdot 2) =$ $(16, 64)$ Всего: 80
			$(8, 32 \cdot 2) =$ $= (8, 64)$ Всего: 72	$(8, 64 \cdot 2) =$ $= (8, 128)$ Всего: 136
	$(7 \cdot 2, 31) =$ $= (14, 31)$ Всего: 45	$(14, 31 \cdot 2) =$ $= (14, 62)$ Всего: 76		
	$(7, 31 \cdot 2) =$ $= (7, 62)$ Всего: 69	$(7, 62 \cdot 2) =$ $= (7, 124)$ Всего: 131		

Ответ:

Задание 1. Ваня выигрывает своим первым ходом.

Задание 2. Петя выигрывает своим вторым ходом.

Задание 3. Ваня выигрывает первым или вторым ходом.

Определение начальной позиции, обеспечивающей выигрыш того или иного игрока

Пример 7.47

Два игрока, Петя и Ваня, играют в следующую игру. Перед игроками лежит куча камней. Игроки ходят по очереди, первый ход делает Петя. За один ход игрок может добавить в кучу один камень или увеличить количество камней в куче в три раза. Например, имея кучу из 15 камней, за один ход

можно получить кучу из 16 или 45 камней. У каждого игрока, чтобы делать ходы, есть неограниченное количество камней.

Игра завершается в тот момент, когда количество камней в куче становится не менее 48. Победителем считается игрок, сделавший последний ход, т.е. первым получивший кучу, в которой будет 48 или больше камней. В начальный момент в куче было S камней, $1 \leq S \leq 47$.

Будем говорить, что игрок имеет *выигрышную стратегию*, если он может выиграть при любых ходах противника. Описать стратегию игрока — значит описать, какой ход он должен сделать в любой ситуации, которая ему может встретиться при различной игре противника.

Выполните следующие задания. Во всех случаях обосновывайте свой ответ.

1. а) Укажите все такие значения числа S , при которых Петя может выиграть в один ход. Обоснуйте, что найдены все нужные значения S , и укажите выигрышающий ход для каждого указанного значения S .

б) Укажите такое значение S , при котором Петя не может выиграть за один ход, но при любом ходе Пети Ваня может выиграть своим первым ходом. Опишите выигрышную стратегию Вани.

2. Укажите два таких значения S , при которых у Пети есть выигрышная стратегия, причём (а) Петя не может выиграть за один ход и (б) Петя может выиграть своим вторым ходом независимо от того, как будет ходить Ваня. Для каждого указанного значения S опишите выигрышную стратегию Пети.

3. Укажите значение S , при котором:

- у Вани есть выигрышная стратегия, позволяющая ему выиграть первым или вторым ходом при любой игре Пети, и
- у Вани нет стратегии, которая позволит ему гарантированно выиграть первым ходом.

Для указанного значения S опишите выигрышную стратегию Вани. Постройте дерево всех партий, возможных при

этой выигрышной стратегии Вани (в виде рисунка или таблицы). На ребрах дерева указывайте, кто делает ход, в узлах — количество камней в куче.

Решение

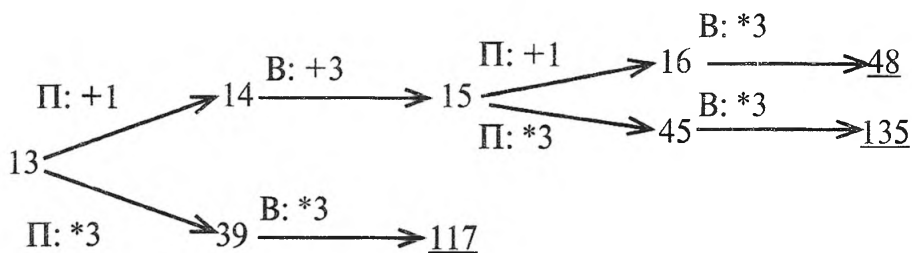
1. а) Петя может выиграть, если 16, ..., 47. Во всех этих случаях достаточно утроить количество камней. При меньших значениях S за один ход нельзя получить кучу, в которой больше 47 камней.

б) Ваня может выиграть первым ходом (как бы ни играл Петя), если исходно в куче будет $S = 15$ камней. Тогда после первого хода Пети в куче будет 16 или 45 камней. В обоих случаях Ваня утраивает количество камней и выигрывает в один ход.

2. Возможные значения S : 5 и 14. В этих случаях Петя, очевидно, не может выиграть первым ходом. Однако он может получить кучу из 15 камней: в первом случае утроением, во втором добавлением одного камня. Эта позиция разобрана в п. 1б. В ней игрок, который будет ходить (теперь это Ваня), выиграть не может, а его противник (т.е. Петя) следующим ходом выиграет.

3. Возможное значение S : 13. После первого хода Пети в куче будет 14 или 39 камней. Если в куче станет 39 камней, Ваня утроит количество камней и выиграет первым ходом. Ситуация, когда в куче 14 камней, уже разобрана в п. 2. В этой ситуации игрок, который будет ходить (теперь это Ваня), выигрывает своим вторым ходом.

На рисунке изображено дерево игры. Выигрышные позиции подчеркнуты.



Ответ:

- а) S от 16 до 47;
б) $S = 15$.
- $S = 5$ и $S = 14$.
- $S = 13$.

Задача № 6**Анализ, исполнение и построение
линейного алгоритма для исполнителя**

Алгоритм — это точно сформулированное исполнителю предписание совершить определенную последовательность действий для решения задачи за конечное число шагов.

Алгоритм может быть задан одним из следующих способов:

- словесное описание последовательности действий на естественном языке;
- графическое изображение в виде блок-схемы;
- запись при помощи псевдокода (алгоритмического языка);
- запись на языке программирования.

В этом типе задач рассматривается в основном словесное описание алгоритмов на естественном языке, а потому никаких специальных знаний для решения задачи не требуется.

**Проверка буквенной последовательности
на соответствие алгоритму****Пример 8.1**

Из букв А, И, З, У, Т, М, К, С формируется слово. Известно, что слово сформировано по следующим правилам:
а) в слове нет подряд идущих двух гласных или двух согласных;

б) первая буква слова в русском алфавите стоит до буквы «К».

Какое из следующих слов удовлетворяет всем перечисленным условиям?

1) АЗИМУТ

2) ТУЗИК

3) МУЗА

4) АИСТ

Решение

Поочередно проанализируем каждое слово:

- 1) а) выполняется б) выполняется (буква «А» в русском алфавите стоит до буквы «К»);
- 2) а) выполняется б) не выполняется (буква «Т» в русском алфавите стоит после буквы «К»);
- 3) а) выполняется б) не выполняется (буква «М» в русском алфавите стоит после буквы «К»);
- 4) а) не выполняется (две подряд идущие гласные).

Ответ: 1.

Поиск числа, соответствующего алгоритму

Пример 8.2

На вход алгоритма подаётся натуральное число N . Алгоритм строит по нему новое число R следующим образом.

1. Строится двоичная запись числа N .

2. К этой записи дописываются справа ещё два разряда по следующему правилу:

а) складываются все цифры двоичной записи, и остаток от деления суммы на 2 дописывается в конец числа (справа). Например, запись 11100 преобразуется в запись 111001;

б) над этой записью производятся те же действия — справа дописывается остаток от деления суммы цифр на 2.

Полученная таким образом запись (в ней на два разряда больше, чем в записи исходного числа N) является двоичной записью искомого числа R .

Укажите такое наименьшее число N , для которого результат работы алгоритма больше 125. В ответе это число запишите в десятичной системе счисления.

Решение

Алгоритм приписывает в конце числа 10, если в двоичной записи числа было нечетное количество единиц, или 00, если четное. Наименьшее число N найдем, если возьмем наименьший результат, больший 125. Это число 126.

$126_{10} = 1111110_2$ может получиться в результате работы алгоритма из числа 11111_2 .

$$11111_2 = 31_{10}.$$

Ответ: 31.

Пример 8.3

Автомат получает на вход трёхзначное число. По этому числу строится новое число по следующим правилам.

1. Складываются первая и вторая, а также вторая и третья цифры исходного числа.
2. Полученные два числа записываются друг за другом в порядке убывания (без разделителей).

Пример. Исходное число: 348. Суммы: $3 + 4 = 7$; $4 + 8 = 12$. Результат: 127. Укажите наименьшее число, в результате обработки которого автомат выдаст число 1412.

Решение

Наименьшим число будет тогда, когда на первом месте стоит наименьшая возможная цифра. Поскольку сумма пер-

вой и второй цифр равна 14 или 12, то наименьшая первая цифра — это 3 (в сумме с 9 даст 12), тогда вторая цифра — это 9. А третья цифра в сумме со второй дает 14, т.е. равна $14 - 9 = 5$.

Получилось число 395.

Ответ: 395.

Пример 8.4

Автомат получает на вход четырёхзначное десятичное число. По этому числу строится новое число по следующим правилам.

1. Складываются первая и вторая, а также третья и четвертая цифры.

2. Полученные два числа записываются друг за другом в порядке возрастания (без разделителей).

Пример. Исходное число: 8754. Суммы: $8 + 7 = 15$; $5 + 4 = 9$. Результат: 915. Определите, сколько из приведенных ниже чисел могут быть получены, как результат работы автомата.

1419 1518 406 911

1) 1 2) 2 3) 3 4) 4

Решение:

Проанализируем поочередно все числа на соответствие алгоритму:

1419 — не соответствует, так как сумма двух цифр не может дать число 19;

1518 — соответствует, например, на вход могло подаваться число 9699;

406 — не соответствует, так как $40 < 6$ (не соблюдается порядок возрастания);

911 — соответствует, например, на вход могло подаваться число 3656.

Итого, может быть получено два числа из списка.

Ответ: 2.

Построение алгоритма минимальной длины

Пример 8.5

У исполнителя Калькулятор две команды, которым присвоены номера:

1. прибавь 2,
2. умножь на 5.

Выполняя первую из них, Калькулятор прибавляет к числу на экране 2, а выполняя вторую, умножает его на 5.

Например, программа 2121 — это программа

умножь на 5,
прибавь 2,
умножь на 5,
прибавь 2,

которая преобразует число 1 в число 37.

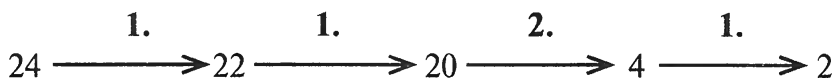
Запишите порядок команд в программе, которая преобразует число 2 в число 24 и содержит не более четырех команд. Указывайте лишь номера команд.

Решение:

Решим обратную задачу — получим из числа 24 число 2. Тогда обозначим команды:

1. вычти 2,
2. раздели на 5.

Запишем цепочку преобразования числа 24 в число 2:



Тогда для преобразования числа 2 в число 24 запишем последовательность команд в обратном направлении: 1211.

Ответ: 1211.



Пример 8.6

У исполнителя Квадр две команды, которым присвоены номера:

1. прибавь 1,
2. возведи в квадрат.

Первая из этих команд увеличивает число на экране на 1, вторая — возводит в квадрат. Программа для исполнителя Квадр — это последовательность номеров команд.

Например, 22111 — это программа

возведи в квадрат
возведи в квадрат
прибавь 1
прибавь 1
прибавь 1

Эта программа преобразует число 3 в число 84.

Запишите программу для исполнителя Квадр, которая преобразует число 5 в число 2500 и содержит не более 6 команд. Если таких программ более одной, то запишите любую из них.

Решение

Решим обратную задачу — получим из числа 2500 число 5. Тогда обозначим команды:

1. вычти 1,
2. извлеки квадратный корень.

Запишем цепочку преобразования числа 2500 в число 5:

2. 1. 2. 1. 1.
 2500 $\longrightarrow$ 50 $\longrightarrow$ 49 $\longrightarrow$ 7 $\longrightarrow$ 6 $\longrightarrow$ 5

Тогда для преобразования числа 5 в число 2500 запишем последовательность команд в обратном направлении: 11212.

Ответ: 11212.

Нестандартные исполнители

Пример 8.7

Исполнитель КУЗНЕЧИК живёт на числовой оси. Начальное положение КУЗНЕЧИКА — точка 0. Система команд Кузнечика:

Вперед 5 — Кузнечик прыгает вперёд на 5 единиц,

Назад 3 — Кузнечик прыгает назад на 3 единицы.

Какое наименьшее количество раз должна встретиться в программе команда «Назад 3», чтобы Кузнечик оказался в точке 21?

Решение

Обозначим через x количество команд «Вперед 5» в программе, а через y — количество команд «Назад 3», причём x и y могут быть только неотрицательными целыми числами.

Для того чтобы КУЗНЕЧИК попал в точку 21 из точки 0, должно выполняться условие: $5 \cdot x - 3 \cdot y = 21$.

Тогда $5 \cdot x = 21 + 3 \cdot y$.

Из последнего уравнения видно, что правая часть должна делиться на 5. Наименьшее y , при котором это условие выполняется, равно 3.

Ответ: 3.

Пример 8.8

Исполнитель Робот ходит по клеткам бесконечной вертикальной клетчатой доски, переходя по одной из команд вверх, вниз, вправо, влево в соседнюю клетку в указанном направлении. Робот выполнил следующую программу:

вверх

влево

влево

вверх
вправо
вверх
вправо

Укажите наименьшее возможное число команд в программе, переводящей Робота из той же начальной клетки в ту же конечную.

Решение

Пары команд (вверх, вниз) и (влево, вправо) возвращают робота в исходное положение. Поэтому можем удалить из списка такие пары. Останется список

вверх
вверх
вверх

То есть всего 3 команды.

Ответ: 3.

Пример 8.9

Исполнитель Чертежник имеет перо, которое можно поднимать, опускать и перемещать. При перемещении опущенного пера за ним остается след в виде прямой линии. У исполнителя существуют следующие команды:

Сместиться на вектор (a, b) — исполнитель перемещается в точку, в которую можно попасть из данной, пройдя a единиц по горизонтали и b — по вертикали. Запись: Повторить 5[Команда 1 Команда 2] означает, что последовательность команд в квадратных скобках повторяется 5 раз.

Чертежник находится в начале координат. Чертежнику дан для исполнения следующий алгоритм:

Сместиться на вектор $(5, 2)$
Сместиться на вектор $(-3, 3)$

Повторить 3[Сместиться на вектор (1,0)]

Сместиться на вектор (3, 1)

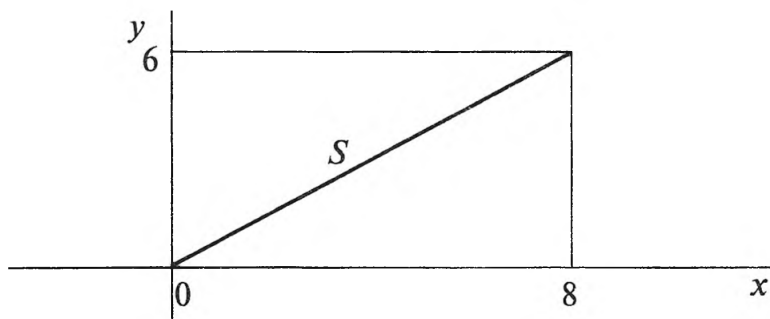
На каком расстоянии от начала координат будет находиться исполнитель Чертежник в результате выполнения данного алгоритма?

Решение

Конечная точка будет обладать координатами по оси x и y . Эти координаты можно складывать независимо друг от друга.

Найдём значение x : $5 - 3 + 1 + 1 + 1 + 3 = 8$.

Найдём значение y : $2 + 3 + 1 = 6$.



По теореме Пифагора $S^2 = X^2 + Y^2 = 100$.

$S = 10$.

Ответ: 10.

Задача № 14

Исполнение алгоритма с фиксированным набором команд

В этой задаче используется в основном описание алгоритмов на псевдокоде (условном алгоритмическом языке, включающем в себя и элементы языка программирования, и элементы обычного естественного языка).

Основные конструкции псевдокода описаны перед текстом задачи.

Исполнитель Чертежник

Пример 8.10

Исполнитель Чертёжник перемещается на координатной плоскости, оставляя след в виде линии. Чертёжник может выполнять команду **сместиться на** (a, b) , где a, b — целые числа. Эта команда перемещает Чертёжника из точки с координатами (x, y) в точку с координатами $(x + a, y + b)$.

Например, если Чертёжник находится в точке с координатами $(4, 2)$, то команда **сместиться на** $(2, -3)$ переместит Чертёжника в точку $(6, -1)$.

Цикл

ПОВТОРИ число РАЗ
 последовательность команд
КОНЕЦ ПОВТОРИ

означает, что *последовательность команд* будет выполнена указанное *число* раз (число должно быть натуральным).

Чертёжнику был дан для исполнения следующий алгоритм (буквами n, a, b обозначены неизвестные числа, $n > 1$):

НАЧАЛО
 сместиться на $(60, 100)$
 ПОВТОРИ n РАЗ
 сместиться на (a, b)
 сместиться на $(33, 44)$
 КОНЕЦ ПОВТОРИ
 сместиться на $(13, 200)$
 сместиться на $(-1, 60)$
КОНЕЦ

Укажите наибольшее возможное значение числа n , для которого найдутся такие значения чисел a и b , что после выполнения программы Чертёжник возвратится в исходную точку.

Решение

В результате выполнения алгоритма Чертежник переместится

по оси x на:

$$60 + n \cdot a + n \cdot 33 + 13 - 1;$$

по оси y на:

$$100 + n \cdot b + n \cdot 44 + 200 + 60.$$

Известно, что в результате перемещения Чертежник вернулся в исходную точку, т.е. перемещение по оси x равно нулю и перемещение по оси y равно нулю:

$$60 + n \cdot a + n \cdot 33 + 13 - 1 = 0;$$

$$100 + n \cdot b + n \cdot 44 + 200 + 60 = 0.$$

То есть

$$n \cdot (a + 33) = -72;$$

$$n \cdot (b + 44) = -360.$$

Наибольшее n — это наибольший общий делитель чисел -72 и -360 . Это число 72.

Ответ: 72.

Пример 8.11

Исполнитель Чертёжник перемещается на координатной плоскости, оставляя след в виде линии. Чертежник может выполнять команду **сместиться на** (a, b) , где a, b — целые числа. Эта команда перемещает Чертежника из точки с координатами (x, y) в точку с координатами $(x + a, y + b)$. Например, если Чертёжник находится в точке с координатами $(4, 2)$, то команда **сместиться на** $(2, -3)$ переместит Чертёжника в точку $(6, -1)$.

Цикл

ПОВТОРИ число РАЗ

последовательность команд

КОНЕЦ ПОВТОРИ

означает, что *последовательность команд* будет выполнена указанное *число раз* (число должно быть натуральным).

Чертёжнику был дан для исполнения следующий алгоритм (количество повторений и смещения в первой из повторяемых команд неизвестны):

НАЧАЛО

сместиться на (4, 6)

ПОВТОРИ ... РАЗ

сместиться на (... , ...)

сместиться на (-1, -2)

КОНЕЦ ПОВТОРИ

сместиться на (20, 30)

КОНЕЦ

После выполнения этого алгоритма Чертежник возвращается в исходную точку. Какое наибольшее число повторений могло быть указано в конструкции «ПОВТОРИ ... РАЗ»?

Решение

Пусть в алгоритме указано ПОВТОРИ x РАЗ и **сместиться на (a, b)** .

В результате выполнения алгоритма Чертежник переместится

по оси x на:

$$4 + x \cdot a - x + 20;$$

по оси y на:

$$6 + x \cdot b - 2 \cdot x + 30.$$

Известно, что в результате перемещения Чертежник вернулся в исходную точку, т.е. перемещение по оси x равно нулю и перемещение по оси y равно нулю:

$$4 + x \cdot a - x + 20 = 0;$$

$$6 + x \cdot b - 2 \cdot x + 30 = 0.$$

То есть

$$x \cdot (a - 1) = -24$$

$$x \cdot (b - 2) = -36$$

x — наибольший общий делитель чисел -24 и -36 (обязательно положительный, т.к. это количество выполнений цикла). Это число 12.

Ответ: 12.

Исполнитель РОБОТ

Пример 8.12

Система команд исполнителя РОБОТ, «живущего» в прямоугольном лабиринте на клетчатой плоскости:

вверх	вниз	влево	вправо
-------	------	-------	--------

При выполнении любой из этих команд РОБОТ перемещается на одну клетку соответственно (по отношению к наблюдателю): вверх $\uparrow$, вниз $\downarrow$, влево $\leftarrow$, вправо $\rightarrow$.

Четыре команды проверяют истинность условия отсутствия стены у каждой стороны той клетки, где находится РОБОТ (также по отношению к наблюдателю):

сверху свободно	снизу свободно	слева свободно	справа свободно
--------------------	-------------------	-------------------	--------------------

Цикл

ПОКА < условие >

последовательность команд

КОНЕЦ ПОКА

выполняется, пока условие истинно.

В конструкции

ЕСЛИ < условие >

ТО команда1
 ИНАЧЕ команда2
 КОНЕЦ ЕСЛИ

выполняется команда1 (если условие истинно) или команда2 (если условие ложно)

Если РОБОТ начнёт движение в сторону находящейся рядом с ним стены, то он разрушится и программа прервётся.

Сколько клеток лабиринта соответствуют требованию, что, начав движение в ней и выполнив предложенную программу, РОБОТ уцелеет и остановится в закрашенной клетке (клетка F6)?

НАЧАЛО

ПОКА снизу свободно ИЛИ справа свободно
 ПОКА справа свободно

вправо

КОНЕЦ ПОКА

вниз

КОНЕЦ ПОКА

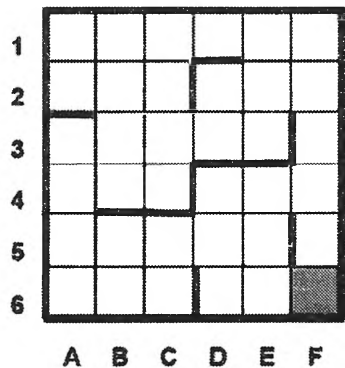
КОНЕЦ

1) 22

2) 19

3) 15

4) 12



Решение

В данной программе РОБОТ сначала проверяет, свободна ли клетка справа или снизу от него. Если это так, то РОБОТ переходит к первому действию внутри цикла. В этом цикле пока у правой стороны клетки, в которой находится РОБОТ, нет стены, он продолжает двигаться вправо. Как только это условие перестанет выполняться, он переходит ко второму

действию внутри цикла. Второе действие заключается в следующем: РОБОТ передвигается на одну клетку вниз. После чего возвращается к началу внешнего цикла.

Проверив последовательно все клетки по правилу движения РОБОТА, выясняем, что число клеток, удовлетворяющих условию задачи, равно 15 (вся первая строка, весь столбец *F*, клетки *D2*, *E2*, *D4*, *D6*, *E4*).

1	✓	✓	✓	✓	✓	✓
2				✓	✓	✓
3						✓
4				✓	✓	✓
5						✓
6						✓
	A	B	C	D	E	F

Правильный ответ указан под номером 3.

Ответ: 3.

Пример 8.13

Система команд исполнителя РОБОТ, «живущего» в прямоугольном лабиринте на клетчатой плоскости, состоит из 8 команд. Четыре команды — это команды-приказы:

вверх	вниз	влево	вправо
-------	------	-------	--------

При выполнении любой из этих команд РОБОТ перемещается на одну клетку соответственно: вверх ↑, вниз ↓, влево ←, вправо →.

Четыре команды проверяют истинность условия отсутствия стены у каждой стороны той клетки, где находится РОБОТ:



сверху свободно	снизу свободно	слева свободно	справа свободно
--------------------	-------------------	-------------------	--------------------

Цикл

ПОКА *условие*

последовательность команд

КОНЕЦ ПОКА

выполняется, пока условие истинно.

В конструкции

ЕСЛИ *условие*

ТО *команда 1*

ИНАЧЕ *команда 2*

КОНЕЦ ЕСЛИ

выполняется команда 1 (если условие истинно) или команда 2 (если условие ложно).

В конструкциях ПОКА и ЕСЛИ условие может содержать команды проверки, а также слова И, ИЛИ, НЕ, обозначающие логические операции.

Если РОБОТ начнет движение в сторону находящейся рядом с ним стены, то он разрушится и программа прервётся.

Сколько клеток лабиринта соответствуют требованию, что, начав движение в ней и выполнив предложенную программу, РОБОТ уцелеет и остановится в закрашенной клетке (клетка F6)?

НАЧАЛО

ПОКА снизу свободно ИЛИ справа свободно

ПОКА справа свободно

ТО

вправо

КОНЕЦ ПОКА

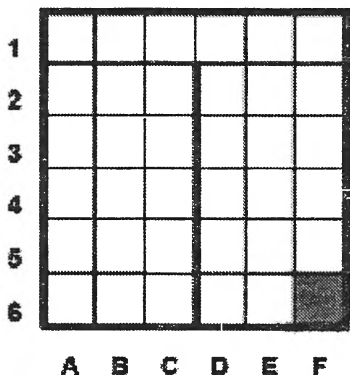
ЕСЛИ снизу свободно

ТО

вниз

КОНЕЦ ЕСЛИ
КОНЕЦ ПОКА
КОНЕЦ

- 1) 18
- 2) 19
- 3) 20
- 4) 21

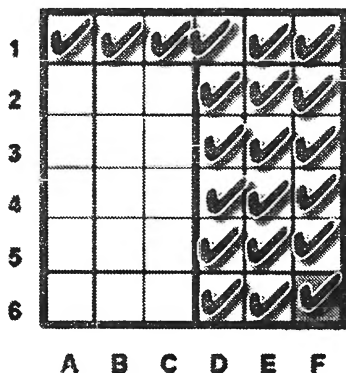


Решение

При данной программе РОБОТ сначала проверяет, свободна ли клетка справа или снизу от него, если это так, то РОБОТ переходит к первому условию. По этому условию, пока справа от клетки, в которой находится РОБОТ, нет стены, он продолжает двигаться вправо. Как только это условие перестанет выполняться, он переходит ко второму условию. Это условие аналогично первому, только теперь проверяется отсутствие стены у нижней стороны клетки.

РОБОТ будет двигаться вправо или вниз, пока у него есть такая возможность, и никогда не разобьется.

Проверив последовательно все клетки по правилу движения РОБОТА, выясняем, что число клеток, удовлетворяющих условию задачи, равно 21.



Правильный ответ указан под номером 4.

Ответ: 4.

Пример 8.14

Система команд исполнителя РОБОТ, «живущего» в клетках прямоугольного лабиринта на плоскости:

вверх, вниз, влево, вправо.

При выполнении любой из этих команд РОБОТ перемещается на одну клетку соответственно: вверх ↑, вниз ↓, влево ←, вправо →.

Четыре условия позволяют проверить отсутствие преград у каждой из сторон той клетки, где находится РОБОТ: **сверху свободно, снизу свободно, слева свободно, справа свободно.**

В цикле

ПОКА <условие> команда

команда выполняется, пока условие истинно, иначе происходит переход на следующую строку программы. Если РОБОТ начнёт движение в сторону стены, рядом с которой он стоит, то робот разрушится и выполнение программы прервётся. Сколько клеток лабиринта соответствуют требованию, что, выполнив предложенную программу, РОБОТ уцелеет и остановится в той же клетке, с которой он начал движение?

НАЧАЛО

ПОКА < снизу свободно > влево

ПОКА < слева свободно > вверх

ПОКА < сверху свободно > вправо

ПОКА < справа свободно > вниз

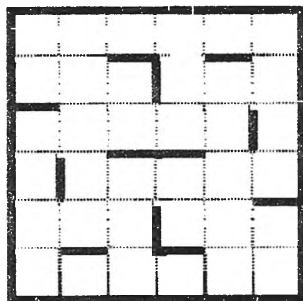
КОНЕЦ

1) 1

2) 2

3) 3

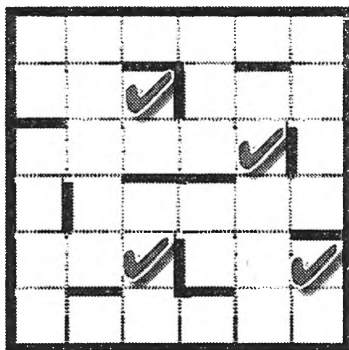
4) 4



Решение

Выясним, что необходимо, чтобы РОБОТ остановился в той же клетке, с которой он начал движение. Так как программа заканчивается командой «ПОКА < справа свободно > вниз», для того чтобы робот остановился в той же клетке, с которой он начал движение, необходимо, чтобы у этой клетки была стенка справа.

Проверим каждую такую клетку. Удовлетворяющими условию задачи будут клетки В2, Д3, В5 и Е5.



Правильный ответ указан под номером: 4.

Ответ: 4

Исполнитель Редактор

Пример 8.15

Исполнитель Редактор получает на вход строку цифр и преобразовывает ее. Редактор может выполнять две команды, в обеих командах v и w обозначают цепочки цифр.

А) **заменить** (v, w).

Эта команда заменяет в строке первое слева вхождение цепочки v на цепочку w . Например, выполнение команды **заменить** (111, 27) преобразует строку 05111150 в строку 0527150. Если в строке нет вхож-

дений цепочки v , то выполнение команды заменить (v, w) не меняет эту строку.

Б) **нашлось** (v).

Эта команда проверяет, встречается ли цепочка v в строке исполнителя Редактор. Если она встречается, то команда возвращает логическое значение «истина», в противном случае возвращает значение «ложь». Строка исполнителя при этом не изменяется.

Цикл

ПОКА условие

последовательность команд

КОНЕЦ ПОКА

выполняется, пока условие истинно.

В конструкции

ЕСЛИ условие

ТО команда1

ИНАЧЕ команда2

КОНЕЦ ЕСЛИ

выполняется команда1 (если условие истинно) или команда2 (если условие ложно).

Какая строка получится в результате применения приведённой ниже программы к строке, состоящей из 68 идущих подряд цифр 8? В ответе запишите полученную строку.

НАЧАЛО

ПОКА нашлось (222) ИЛИ нашлось (888)

ЕСЛИ нашлось (222)

ТО заменить (222, 8)

ИНАЧЕ заменить (888, 2)

КОНЕЦ ЕСЛИ

КОНЕЦ ПОКА

КОНЕЦ

Решение

Обозначим строку из 68 восьмерок — $68\langle 8 \rangle$, строку из двойки и 65 восьмерок — $1\langle 2 \rangle 65\langle 8 \rangle$ и т.д.

Отработаем 4 первых цикла программы:

$68\langle 8 \rangle \rightarrow 1\langle 2 \rangle 65\langle 8 \rangle \rightarrow 2\langle 2 \rangle 62\langle 8 \rangle \rightarrow 3\langle 2 \rangle 59\langle 8 \rangle \rightarrow 60\langle 8 \rangle$

В результате количество восьмерок уменьшилось на 8. Не сложно понять, что строка будет уменьшаться на 8 восьмерок каждые 4 итерации. В результате останется строка из 4 восьмерок. Доработаем программу:

$\dots \rightarrow 4\langle 8 \rangle \rightarrow 1\langle 2 \rangle 1\langle 8 \rangle = 28$.

Ответ: 28.

Пример 8.16

Имеется фрагмент алгоритма, записанный на алгоритмическом языке:

```

m := 10
b := Извлечь(a, m)
нц для k от 4 до 5
    c := Извлечь(a, k)
    b := Склеить(b, c)
кц
нц для k от 1 до 3
    c := Извлечь(a, k)
    b := Склеить(b, c)
кц
    
```

Здесь переменные **a**, **b** и **c** — строкового типа; переменные **n**, **m**, **k** — целые. В алгоритме используются следующие функции:

Извлечь(**x**, **i**) — возвращает **i**-й символ слева в строке **x**. Имеет строковый тип.

Склеить(**x**, **y**) — возвращает строку, в которой записаны подряд сначала все символы строки **x**, а затем все

символы строки **y**. Имеет строковый тип. Значения строк записываются в кавычках (одинарных), например **x='школа'**.

Какое значение примет переменная **b** после выполнения этого фрагмента алгоритма,

если переменная **a** имела значение 'ИНФОРМАТИКА'?

- 1) 'ФОРМАТ' 2) 'ФОРИНТ'
- 3) 'КОРТИК' 4) 'КОРИНФ'

Решение

Отработаем последовательно алгоритм:

m := 10

b := Извлечь(a, m) **b := Извлечь('ИНФОРМАТИКА', 10) = 'К'**

нц для k от 4 до 5

c := Извлечь(a, k) **c := Извлечь('ИНФОРМАТИКА', 4) = 'О'**

b := Склеить(b, c) **b := Склеить('К', 'О') = 'КО'**

c := Извлечь('ИНФОРМАТИКА', 5) = 'Р'

b := Склеить('КО', 'Р') = 'КОР'

кц

нц для k от 1 до 3

c := Извлечь(a, k) **c := Извлечь('ИНФОРМАТИКА', 1) = 'И'**

b := Склеить(b, c) **b := Склеить('КОР', 'И') = 'КОРИ'**

c := Извлечь('ИНФОРМАТИКА', 2) = 'Н'

b := Склеить('КОРИ', 'Н') = 'КОРИН'

c := Извлечь('ИНФОРМАТИКА', 3) = 'Ф'

b := Склеить('КОРИН', 'Ф') = 'КОРИНФ'

кц

В результате выполнения алгоритма переменная **b** приняла значение 'КОРИНФ'.

Ответ: 4.

Исполнитель Черепашка

Пример 8.17

Исполнитель Черепашка перемещается на экране компьютера, оставляя след в виде линии. В каждый



конкретный момент известно положение исполнителя и направление его движения. У исполнителя существуют две команды:

Вперед n , где n — целое число, вызывающее передвижение черепашки на n шагов в направлении движения.

Направо m , где m — целое число, вызывающее изменение направления движения на m градусов по часовой стрелке.

Запись **Повтори 5 [Команда1 Команда2]** означает, что последовательность команд в скобках повторится 5 раз.

Черепашке был дан для исполнения следующий алгоритм:

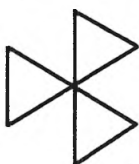
Повтори 5 [Повтори 4 [Вперед 40 Направо 90] Направо 120]

Какая фигура появится на экране?

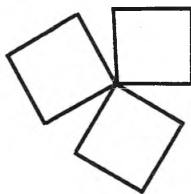
1)



2)



3)



4)



Решение

Последовательность действий **Вперед 40 Направо 90** рисует отрезок длиной 40 шагов, а затем меняет направление на 90 градусов по часовой стрелке. Тогда последовательность **Повтори 4 [Вперед 40 Направо 90]** нарисует квадрат, а направление вернется в исходное.

Затем выполняется команда **Направо 120**, она изменит направление на 120 градусов от исходного.

Если повторить все рассмотренные действия 5 раз:

Повтори 5 [Повтори 4 [Вперед 40 Направо 90] Направо 120], то будет 5 раз нарисован квадрат. Причем каждый следующий повернут вокруг вершины относительно предыдущего на 120 градусов. Не сложно заметить, что 4-й квадрат будет нарисован поверх первого ($120 \cdot 3 = 360$, сделан поворот на целый круг, возврат в исходное положение), а 5-й поверх второго.

Результат изображен под номером 3.

Ответ: 3.

Задача № 20

Анализ алгоритма, содержащего цикл и ветвление

В этой задаче используется описание алгоритмов на языках программирования.

Основными используемыми конструкциями являются конструкции ветвления и циклов.

Конструкция ветвления в языке Паскаль

Полная	Неполная
if <логическое выражение> then <оператор1> else <оператор2>;	if <логическое выражение> then <оператор1>;

При этом оператор может быть простым, а может быть составным.

Составной оператор имеет структуру:

begin

<Последовательность_операторов>

end

Конструкция циклов в языке Паскаль

Цикл со счетчиком (или цикл **for**)

Этот цикл используется, когда число повторений не связано с тем, что происходит в теле цикла. То есть количество повторений может быть вычислено заранее.

В заголовке цикла указываются два значения. Первое значение присваивается так называемой переменной-счетчику, от этого значения начинается отсчет количества итераций (повторений). Отсчет идет всегда с шагом, равным единице. Второе значение указывает, при каком значении счетчика цикл должен остановиться. Другими словами, количество итераций цикла определяется разностью между вторым и первым значением плюс единица. В Паскале тело цикла не должно содержать выражений, изменяющих счетчик.

Цикл **for** существует в двух формах:

```
for<счетчик>:=<значение>to<конечное_значение>do  
    <тело_цикла>;
```

```
for <счетчик>:=<значение> downto <конечное_значение> do  
    <тело_цикла>;
```

Счетчик — это переменная любого из перечисляемых типов (целого, булевого, символьного, диапазонного, перечисления). Начальные и конечные значения могут быть представлены не только значениями, но и выражениями, возвращающими совместимые с типом счетчика типы данных. Если между начальным и конечным выражением указано служебное слово **to**, то на каждом шаге цикла значение параметра будет увеличиваться на единицу. Если же указано **downto**, то значение параметра будет уменьшаться на единицу.

Цикл с предусловием (или цикл **while**)

Цикл **while** является циклом с предусловием. В заголовке цикла находится логическое выражение. Если оно возвращает **true**, то тело цикла выполняется, если **false** — то нет.

```
while<логическое_выражение>do  
<тело_цикла>;
```

Когда тело цикла было выполнено, то ход программы снова возвращается в заголовок цикла. Условие выполнения тела снова проверяется (находится значение логического выражения). Тело цикла выполнится столько раз, сколько раз логическое выражение вернет **true**. Поэтому очень важно в теле цикла предусмотреть изменение переменной, фигурирующей в заголовке цикла, таким образом, чтобы когда-нибудь обязательно наступала ситуация **false**. Иначе произойдет заикливание.

Цикл с постусловием (или цикл **repeat**)

Цикл **while** может не выполниться ни разу, если логическое выражение в заголовке сразу вернуло **false**. Однако бывают ситуации, когда тело цикла должно выполниться хотя бы один раз, независимо оттого, что вернет логическое выражение. В таком случае используется цикл **repeat** — цикл с постусловием.

В цикле **repeat** логическое выражение стоит после тела цикла. В случае **true** происходит выход из цикла, в случае **false** — его повторение:

```
repeat<тело_цикла>  
until<логическое_выражение>;
```

Тело цикла может включать в себя другой цикл. Количество вложенных циклов не ограничено.

Операции div и mod

Операция **div** возвращает целочисленный результат деления. При этом дробная часть будет отброшена. $X \text{ div } Y$ вернет целую часть от деления X на Y .

Например, после работы оператора $z := 55 \text{ div } 6$, z будет равно 9.

Операция **mod** возвращает остаток от деления. $X \text{ mod } Y$ вернет остаток от деления X на Y .

Например, после работы оператора $z := 55 \text{ mod } 6$, z будет равно 1.

Поиск наименьшего числа

Пример 8.18

Ниже на пяти языках программирования записан алгоритм. Получив на вход число x , этот алгоритм печатает числа: a и b . Укажите наименьшее положительное пятизначное число x , при вводе которого алгоритм печатает сначала 6, а потом 3.

Бейсик	Python
DIM X, Y, A, B AS INTEGER A = 0 B = 10 INPUT X WHILE X > 0 Y = X MOD 10 X = X \ 10 IF Y > A THEN A = Y IF Y < B THEN B = Y WEND PRINT A PRINT B	a = 0 b = 10 x = int(input()) while x > 0: y = x % 10 x = x // 10 if y > a: a = y if y < b: b = y print(a) print(b)

Паскаль	Алгоритмический язык
<pre> var x, y, a, b: integer; begin a := 0; b := 10; readln(x); while x > 0 do begin y := x mod 10; x := x div 10 if y > a then a := y; if y < b then b := y; end; writeln(a); writeln(b) end.</pre>	<pre> алг нач цел x, y, a, b a := 0 b := 10 ввод x нцпока x > 0 y := mod(x, 10) x := div(x, 10) если y > a то a := y все если y < b то b := y все кц вывод a, нс, b кон</pre>
Сн	
<pre> #include <stdio.h> int main() { int x, y, a, b; a = 0; b = 10; scanf("%d", &x); while (x > 0) { y = x % 10; x = x / 10; if (y > a) a = y; if (y < b) b = y; } printf("%d\n%d\n", a, b); }</pre>	

Решение

Так как оператор **div** возвращает целую часть от деления, то при делении на 10 это равносильно отсечению последней цифры. Значит, цикл выполнится пять раз.

В переменную *a* записывается остаток от деления числа *x* на 10 при условии, что этот остаток больше числа уже записанного в переменной *a*. В переменную *b* записывается остаток от деления числа *x* на 10 при условии, что этот остаток меньше числа уже записанного в переменной *b*. Таким образом, в переменную *a* записывается наибольшая цифра числа *x*, а в переменную *b* — наименьшая. Итак, нужно найти наименьшее пятизначное число, у которого наименьшая цифра равна 3, а наибольшая — равна 6. Такое число: 33336.

Ответ: 33336.

Пример 8.19

Ниже на пяти языках записан алгоритм. Получив на вход число *x*, этот алгоритм печатает два числа *a* и *b*. Укажите наименьшее из таких чисел *x*, при вводе которого алгоритм печатает сначала 2, а потом 13.

Бейсик	Python
DIM X, A, B AS INTEGER INPUT X A = 0: B = 0 WHILE X > 0 A = A + 1 B = B + (X MOD 100) X = X / 100 WEND PRINT A PRINT B	<pre> x = int(input()) a, b = 0, 0 while x > 0: a = a + 1 b = b + x%100 x = x/100 print(a) print(b) </pre>

Паскаль	Алгоритмический язык
<pre> var x, a, b: integer; begin readln(x); a := 0; b := 0; while x > 0 do begin a := a+1; b := b+(x mod 100); x := x div 100; end; writeln(a); write(b); end.</pre>	<pre> алг нач цел x, a, b ввод x a:=0; b:=0 нцпока x > 0 a := a+1 b := b+mod(x,100) x := div(x,100) кц вывод a, nc, b кон</pre>
С++	
<pre> #include <stdio.h> void main() { int x, a, b; scanf("%d", &x); a = 0; b = 0; while (x > 0) { a = a+1; b = b + (x%100); x = x/100; } printf("%d\n%d", a, b); }</pre>	

Решение

Так как оператор **div** возвращает целую часть от деления, то при делении на 100 это равносильно отсечению последних двух цифр.

При каждом выполнении тела цикла от десятичной записи x отсекается две последних цифры до тех пор, пока все цифры не будут отсечены, т.е. x не станет равно 0. В переменной a сохраняется количество выполнений тела цикла. Для того чтобы a стало равным 2, x должно быть трехзначным или четырехзначным.

Оператор **mod** возвращает остаток от деления, при делении на 100 это последние две цифры x . В переменной b суммируются остатки от деления исходного числа на 100. Нужно найти наименьшее число, значит, оно трехзначное. $13 =$ сумма числа, составленного из двух последних цифр, и первой цифры числа. Наименьшим таким числом является 112.

Ответ: 112

Пример 8.20

Ниже на пяти языках программирования записан алгоритм. Получив на вход число x , этот алгоритм печатает число M . Известно, что $x > 100$. Укажите наименьшее такое (т.е. большее 100) число x , при вводе которого алгоритм печатает 26.

Бейсик	Python
<pre> DIM X, L, M AS INTEGER INPUT X L = X M = 65 IF L MOD 2 = 0 THEN M = 53 ENDIF WHILE L <> M IF L > M THEN L = L - M ELSE M = M - L ENDIF WEND PRINT M </pre>	<pre> x = int(input()) L = X M = 65 if L % 2 == 0: M = 52 while L != M: if L > M: L = L - M else: M = M - L print(M) </pre>

Алгоритмический язык	Паскаль
алг нач цел x, L, M ввод x $L := x$ $M := 65$ если $\text{mod}(L, 2) = 0$ то $M := 52$ все нц пока $L < > M$ если $L > M$ то $L := L - M$ иначе $M := M - L$ все кц вывод M кон	var x, L, M : integer; begin readln(x); $L := x$; $M := 65$; if $L \bmod 2 = 0$ then $M := 52$ while $L < > M$ do if $L > M$ then $L := L - M$ else $M := M - L$; writeln(M); end.

Решение

В теле цикла числа M и L уменьшаются, пока не станут равными. Поскольку в итоге напечатано 26, значит, в конце программы оба числа равны 26. Рассмотрим изменение этой пары чисел в обратном порядке (при этом не знаем, какое именно число M , а какое L):

$(26; 26) \rightarrow (52; 26) \rightarrow (78; 52) \rightarrow (130; 52)$.

Если 130 — это L в начале работы (равное x), то так как оно четное, M в начале работы должно быть 52. Условие выполнено для пары $(130; 52)$, значит, искомое число 130.

Ответ: 130.

Пример 8.21

Ниже на четырех языках записан алгоритм. Получив на вход число x , этот алгоритм печатает два числа: a и b . Укажите наименьшее из таких чисел x , при вводе которых алгоритм печатает сначала 2, а потом 21.

Бейсик	Паскаль
<pre> DIM X, A, B AS INTEGER INPUT X A=0: B=1 WHILE X > 0 A = A+1 B = B*(X MOD 10) X = X \ 10 WEND PRINT A PRINT B </pre>	<pre> var x, a, b: integer; begin readln(x); a:=0; b:=1; while x>0 do begin a:=a+1; b:=b*(x mod 10); x:= x div 10 end; writeln(a); write(b); end. </pre>
Сп	Алгоритмический язык
<pre> #include<stdio.h> void main() { int x, a, b; scanf("%d", &x); a=0; b=1; while (x>0){ a=a+1; b=b*(x%10); x= x/10; } printf("%d\n%d", a, b); } </pre>	<pre> алг нач цел x, a, b ввод x a:=0; b:=1 нцпока x>0 a:=a+1 b:=b*mod(x,10) x:=div(x,10) кц вывод а, нс, b кон </pre>

Решение

На каждом шаге цикла от числа отсекается правая цифра, пока число не станет равно нулю. То есть цикл выполняется столько раз, сколько цифр в числе. Это количество

накапливается в переменной a . В конце программы $A = 2$, значит, исходное число двузначное.

В переменной b накапливается произведение остатков от деления на 10, т.е. произведение цифр исходного числа.

Значит, нам нужно найти наименьшее двузначное число, произведение цифр которого равно 21. Такое число 37.

Ответ: 37

Поиск наибольшего числа

Пример 8.22

Ниже записана программа. Получив на вход число x , эта программа печатает два числа, L и M . Укажите наибольшее из таких чисел x , при вводе которых алгоритм печатает сначала 3, а потом 7.

```
var x, L, M: integer;  
begin  
  readln(x);  
  L:=0; M:=0;  
  while x > 0 do begin  
    L:= L + 1;  
    if x mod 2 = 0 then  
      M:= M + (x mod 10) div 2;  
    x:= x div 10;  
  end;  
  writeln(L); write(M);  
end.
```

Решение

На каждом шаге от десятичной записи x отсекается последняя цифра до тех пор, пока все цифры не будут отсечены, т.е. x не станет равно 0. В переменной L накапливается число выполнений цикла. $L = 3$, значит цикл выполнялся 3 раза, значит, исходное число x трехзначное.

Теперь рассмотрим оператор изменения M :

if $x \bmod 2 = 0$ then

$M := M + (x \bmod 10) \div 2$;

end;

M увеличивается только в случае, если текущее значение x четное, т.е. если четная его последняя цифра. В этом случае к текущему значению M прибавляется целочисленный результат деления последней цифры текущего значения x на 2.

Таким образом, M — это сумма половин четных цифр числа x .

Значит, нам нужно найти наибольшее трехзначное число x , сумма четных цифр которого равна 14.

Наибольшее такое число 986.

Ответ: 986.

Пример 8.23

Ниже на пяти языках записан алгоритм. Получив на вход число x , этот алгоритм печатает два числа a и b . Укажите наибольшее из таких чисел x , при вводе которого алгоритм печатает сначала 2, а потом 5.

Бейсик	Python
DIM X, A, B AS INTEGER INPUT X A = 0: B = 1 WHILE X > 0 A = A + 1 B = B * (X MOD 100) X = X \ 100 WEND PRINT A PRINT B	x = int(input()) a, b = 0, 1 while x > 0: a = a + 1 b = b * x % 100 x = x // 100 print(a) print(b)

Паскаль	Алгоритмический язык
<pre> var x, a, b: integer; begin readln(x); a := 0; b := 1; while x > 0 do begin a := a+1; b := b*(x mod 100); x := x div 100; end; writeln(a); write(b); end.</pre>	<pre> алг нач цел x, a, b ввод x a:=0; b:=1 нцпока x > 0 a := a+1 b := b*mod(x,100) x := div(x,100) кц вывод a, nc, b кон</pre>
Си	
<pre> #include <stdio.h> void main() { int x, a, b; scanf("%d", &x); a = 0; b = 1; while (x > 0) { a = a+1; b = b * (x%100); x = x/100; } printf("%d\n%d", a, b); }</pre>	

Решение

Так как оператор **div** возвращает целую часть от деления, то при делении на 100 это равносильно отсечению последних двух цифр.

На каждом шаге от десятичной записи x отсекается две последних цифры до тех пор, пока все цифры не будут отсечены, т.е. x не станет равно 0.

В переменной a накапливается количество выполнений цикла, $a = 2$, значит, число x трехзначное или четырехзначное.

В переменной b накапливается произведение двух чисел: числа, состоящего из последних двух цифр числа x , и первой цифры числа x , если число x трехзначное (или двух первых цифр, если число x четырехзначное).

Так как $b = 5$, то второй из множителей не может быть двузначным, т.е. число x трехзначное. Наибольшее такое число равно 501.

Ответ: 501.

Пример 8.24

Ниже на пяти языках программирования записан алгоритм. Получив на вход число x , этот алгоритм печатает числа: a и b .

Укажите наибольшее положительное пятизначное число x , при котором после выполнения алгоритма будет напечатано сначала 3, а потом 4.

Бейсик	Python
DIM X, Y, A, B AS INTEGER A = 0 B = 0 INPUT X WHILE X > 0 Y = X MOD 10 IF Y > 4 THEN A = A + 1 IF Y < 7 THEN B = B + 1 X = X \ 10 WEND PRINT A PRINT B	a = 0 b = 0 x = int(input()) while x > 0: y = x % 10 if y > 4: a = a + 1 if y < 7: b = b + 1 x = x // 10 print(a) print(b)

Алгоритмический язык	Паскаль
алг нач цел x, y, a, b a := 0 b := 0 ввод x нцпока x > 0 y := mod(x, 10) если y > 4 то a := a + 1 все если y < 7 то b := b + 1 все x := div(x, 10) кц вывод a, nc, b кон	var x, y, a, b: integer; begin a := 0; b := 0; readln(x); while x > 0 do begin y := x mod 10; if y > 4 then a := a + 1; if y < 7 then b := b + 1; x := x div 10 end ; writeln(a); writeln(b) end.
Си	
<pre>#include <stdio.h> int main() { int x, y, a, b; a = 0; b = 0; scanf("%d", &x); while (x > 0) { y = x % 10; if (y > 4) a = a + 1; if (y < 7) b = b + 1; x = x / 10; } printf("%d\n%d\n", a, b); }</pre>	

Решение

a — количество цифр в числе, больших, чем 4. b — количество цифр в числе, меньших, чем 7.

Так как $a = 3$ и $b = 4$, то в исходном числе одна цифра не меньше 7, две цифры не больше, чем 4, и еще две от 5 до 6.

Наибольшее подходящее число равно 96644.

Ответ: 96644.

Задача № 22

Анализ результата исполнения алгоритма

Поиск количества программ по результату

Задачу такого типа можно решить, построив подробное дерево всех возможных вариантов наборов команд и подсчитав те, которые приведут к нужному результату. Однако это очень длинный и объемный способ. Его использование может привести к вычислительным ошибкам, а при большой длине программы построение дерева вообще практически невозможно.

Рассмотрим более эффективный метод подсчета количества программ.

Пример 8.25

У исполнителя Увеличитель две команды, которым присвоены номера:

1. прибавь 2,

2. умножь на 3.

Первая из них увеличивает число на экране на 2, вторая — умножает его на 3.

Программа для Увеличителя — это последовательность команд. Сколько есть программ, которые число 1 преобразуют в число 31?

Ответ обоснуйте.

Решение

Заполним таблицу со следующими столбцами:

«Число» — перечень всех чисел от 1 до 31;

«Числа-источники» — числа, из которых одной из указанных команд можно получить текущее число;

«Количество способов» — количество способов, которыми можно получить текущее число из чисел-источников. Равно сумме значений «Кол-во способов» чисел-источников.

Заметим, что никаким набором указанных команд невозможно получить четное число из 1, значит, четные числа можем в таблице не рассматривать вообще.

Число	Числа-источники	Кол-во способов получения
1	1	1
3	1	2
5	3	2
7	5	2
9	3 ; 7	$2 + 2 = 4$
11	9	4
13	11	4
15	5 ; 13	$2 + 4 = 6$
17	15	6
19	17	6
21	7 ; 19	$2 + 6 = 8$
23	21	8
25	23	8
27	9 ; 25	$4 + 8 = 12$
29	27	12
31	29	12

Число 1 нам дано, т.е. его можно получить единственным способом: ничего не делая.

Число 3 можно получить из 1 двумя способами: командой 1 и командой 2. И т.д.

Заметим, что два источника могут быть только у чисел, кратных трем. Это наблюдение поможет нам быстро заполнить таблицу, т.к. количество способов увеличивается, когда рассматриваем число, кратное трем.

Например, число 9 можно получить из числа 3 и числа 7. Сложив количество способов получения чисел 3 и 7 ($2 + 2 = 4$), получим количество способов получения числа 9.

Полная таблица приведена для наглядности, можно было заполнить только строки с числами, кратными 3, так как только они меняют количество способов получения числа.

Для числа 31 получаем количество способов 12. Это и есть искомое количество программ.

Ответ: 12.

Пример 8.26

У исполнителя Арифметик две команды, которым присвоены номера:

1. прибавь 1,

2. прибавь 3.

Первая из них увеличивает на 1 число на экране, вторая увеличивает это число на 3.

Программа для Арифметика — это последовательность команд.

Сколько существует программ, которые число 2 преобразуют в число 15?

Решение

Заполним таблицу со следующими строками:

«Число» — перечень всех чисел от 2 до 15;

«Числа-источники» — числа, из которых одной из указанных команд можно получить текущее число;

«Количество способов» — количество способов, которыми можно получить текущее число из чисел-источников. Равно сумме значений «Кол-во способов» чисел-источников.

Число	Числа-источники	Кол-во способов получения
2	2	1
3	2	1
4	3	1
5	2 ; 4	$1 + 1 = 2$
6	3 ; 5	$1 + 2 = 3$
7	4 ; 6	$1 + 3 = 4$
8	5 ; 7	$2 + 4 = 6$
9	6 ; 8	$3 + 6 = 9$
10	7 ; 9	$4 + 9 = 13$
11	8 ; 10	$6 + 13 = 19$
12	9 ; 11	$9 + 19 = 28$
13	10 ; 12	$13 + 28 = 41$
14	11 ; 13	$19 + 41 = 60$
15	12 ; 14	$28 + 60 = 88$

Для числа 15 получаем количество способов 88. Это и есть искомое количество программ.

Ответ: 88.

Пример 8.27

Исполнитель Май15 преобразует число на экране. У исполнителя есть две команды, которым присвоены номера:

1. Прибавить 1

2. Умножить на 2

Первая команда увеличивает число на экране на 1, вторая умножает его на 2.

Программа для исполнителя Май15 — это последовательность команд. Сколько существует программ, для которых при исходном числе 2 результатом является число 29 и при этом траектория вычислений содержит число 14 и не содержит числа 25? Траектория вычислений программы — это последовательность результатов выполнения всех команд программы. Например, для программы **121** при исходном числе 7 траектория будет состоять из чисел 8, 16, 17.

Решение

Заполним таблицу со следующими строками:

«Число» — перечень всех чисел от 2 до 29;

«Числа-источники» — числа, из которых одной из указанных команд можно получить текущее число;

«Количество способов» — количество способов, которыми можно получить текущее число из чисел-источников. Равно сумме значений «Кол-во способов» чисел-источников.

Число	Числа-источники	Кол-во способов получения
2	2	1
3	2	1
4	2 ; 3	$1 + 1 = 2$
5	4	2
6	3 ; 5	$1 + 2 = 3$
7	6	3
8	4 ; 7	$2 + 3 = 5$
9	8	5
10	5 ; 9	$2 + 5 = 7$
11	10	7
12	6 ; 11	$3 + 7 = 10$
13	12	10
14	7 ; 13	$3 + 10 = 13$



Число	Числа-источники	Кол-во способов получения
15	14	13
16	15	13
17	16	13
18	17	13
19	18	13
20	19	13
21	20	13
22	21	13
23	22	13
24	23	13
25		
26		
27		
28	14	13
29	28	13

До строки с числом 15 считаем все способы получения чисел. Начиная с числа 16 и до числа 24 (включительно) числа-источники 8-12 и способ получения числа применением команды «Умножить на 2» не подходят, т.к. траектория вычисления не содержит число 14. Для этих чисел учитываем одно число-источник (предыдущее) и один способ получения — «Прибавить 1».

Числа 25 вообще не должно быть в траектории, а значит, числа 26 и 27 получить невозможно.

Для числа 28 существует одно число-источник (14).

Для числа 29 получаем количество способов 13. Это и есть искомое количество программ.

Ответ: 13.

Пример 8.28

У исполнителя Удвоитель-Утроитель три команды, которым присвоены номера:



1. прибавь 1,
2. умножь на 2,
3. умножь на 3.

Первая из них увеличивает на 1 число на экране, вторая увеличивает это число в 2 раза, третья — в 3 раза. Программа для Удвоителя-Утроителя — это последовательность команд. Сколько существует программ, которые число 1 преобразуют в число 14?

Решение

Заполним таблицу со следующими строками:

«Число» — перечень всех чисел от 1 до 14;

«Числа-источники» — числа, из которых одной из указанных команд можно получить текущее число;

«Количество способов» — количество способов, которыми можно получить текущее число из чисел-источников. Равно сумме значений «Кол-во способов» чисел-источников.

Число	Числа-источники	Кол-во способов получения
1	1	1
2	1	2
3	1 ; 2	$1 + 2 = 3$
4	2 ; 3	$2 + 3 = 5$
5	4	5
6	2 ; 3 ; 5	$2 + 3 + 5 = 10$
7	6	10
8	4 ; 7	$5 + 10 = 15$
9	3 ; 8	$3 + 15 = 18$
10	5 ; 9	$5 + 18 = 23$
11	10	23
12	4 ; 6 ; 11	$5 + 10 + 23 = 38$
13	12	38
14	7 ; 13	$10 + 38 = 48$

Для числа 14 получаем количество способов 48. Это и есть искомое количество программ.

Ответ: 48.

Пример 8.29

У исполнителя четыре команды, которым присвоены номера:

1. прибавь 1,
2. сделай чётное,
3. сделай нечётное,
4. умножь на 10.

Первая из них увеличивает на 1 исходное число x , вторая умножает это число на 2, третья переводит число x в число $2x + 1$, четвёртая умножает его на 10. Например, вторая команда переводит число 10 в число 20, а третья переводит число 10 в число 21. Программа для исполнителя — это последовательность команд.

Сколько существует программ, которые число 1 преобразуют в число 15?

Решение

Заполним таблицу со следующими строками:

«Число» — перечень всех чисел от 1 до 15;

«Числа-источники» — числа, из которых одной из указанных команд можно получить текущее число;

«Количество способов» — количество способов, которыми можно получить текущее число из чисел-источников. Равно сумме значений «Кол-во способов» чисел-источников.

Число	Числа-источники	Кол-во способов получения
1	1	1
2	1 ; 1	2
3	1 ; 2	$1 + 2 = 3$

Число	Числа-источники	Кол-во способов получения
4	2 ; 3	$2 + 3 = 5$
5	2 ; 4	$2 + 5 = 7$
6	3 ; 5	$3 + 7 = 10$
7	3 ; 6	$3 + 10 = 13$
8	4 ; 7	$5 + 13 = 18$
9	4 ; 8	$5 + 18 = 23$
10	1 ; 5 ; 9	$1 + 7 + 23 = 31$
11	5 ; 10	$7 + 31 = 38$
12	6 ; 11	$10 + 38 = 48$
13	6 ; 12	$10 + 48 = 58$
14	7 ; 13	$13 + 58 = 71$
15	7 ; 14	$13 + 71 = 84$

Для числа 15 получаем количество способов 84. Это и есть искомое количество программ.

Ответ: 84.

Поиск количества результирующих чисел по количеству команд

Пример 8.30

У исполнителя Калькулятор две команды:

1. умножь на 2,

2. умножь на 3.

Первая из них умножает число на экране на 2, вторая — утраивает его. Сколько различных чисел можно получить из числа 2 с помощью программы, которая содержит ровно 3 команды?

Решение

От перестановки операций умножения результат не изменится, т.е. программы 1.2. и 2.1. дадут один и тот же результат. Значит, нужно посчитать все возможные наборы команд 1. и 2. без учета порядка их выполнения. Такие наборы:

- 3 команды 1.;
- 2 команды 1. + 1 команда 2.;
- 1 команда 1. + 2 команды 2.;
- 3 команды 2.

Значит, количество различных чисел, которые можно получить с помощью программ из 3 команд, равно 4.

Ответ: 4.

Пример 8.31

У исполнителя Калькулятор две команды:

1. прибавь 4,

2. вычти 3.

Первая из них увеличивает число на экране на 4, вторая — уменьшает его на 3 (отрицательные числа допускаются). Программа для Калькулятора — это последовательность команд. Сколько различных чисел можно получить из числа 1 с помощью программы, которая содержит ровно 7 команд?

Решение

От перестановки операций сложения результат не изменится, т.е. программы 1.2. и 2.1. дадут один и тот же результат. Значит, нужно посчитать все возможные наборы длиной 7 команд 1. и 2. без учета порядка выполнения. Такие наборы:

- 7 команд 1.;
- 6 команд 1. + 1 команда 2.;
- 5 команд 1. + 2 команды 2.;
- 4 команды 1. + 3 команды 2.;
- 3 команды 1. + 4 команды 2.;
- 2 команды 1. + 5 команд 2.;
- 1 команда 1. + 6 команд 2.;
- 7 команд 2.

Всего 8 наборов. То есть можно получить 8 различных чисел.

Ответ: 8.

•

1

52

Посчитаем все возможные числа в узлах дерева, пропускающая повторяющиеся.

Всего различных чисел 23.

Ответ: 23.

Задача № 25

Алгоритмы обработки массивов

В задачах этого типа встречаются следующие основные алгоритмические (программные) блоки:

- **Перебор элементов массива**

Для одномерного массива запускается цикл по всем его элементам.

Для двумерного массива — цикл в цикле по строкам и столбцам.

Для n -мерного массива — n вложенных циклов.

- **Перебор элементов массива, удовлетворяющих условию**

Для перебора элементов массива, удовлетворяющих условию (например, только четных), в начале цикла устанавливается условие. Тогда все дальнейшие действия выполняются только при выполнении этого условия. Если же условие не выполняется, цикл переходит к следующему элементу массива.

- **Обмен местами элементов массива**

Производится аналогично обмену значениями переменных. Если нужно поменять местами n -й и k -й элементы массива, то n -й элемент сохраняется в буферную переменную, затем значение k -го элемента копируется в n -й, а значение буферной переменной копируется в k -й элемент.

- **Поиск максимального (минимального) элемента массива**

При поиске максимального элемента массива за предполагаемый максимум берется первый элемент массива или число, заведомо меньшее любого элемента массива. Затем каждый следующий элемент массива сравнивается с предполагаемым максимумом, и если значение этого элемента больше, то оно становится новым предполагаемым максимумом.

Аналогично, при поиске минимального элемента массива за предполагаемый минимум берется первый элемент массива или число, заведомо большее любого элемента массива. Затем каждый следующий элемент массива сравнивается с предполагаемым минимумом, и если значение этого элемента меньше, то оно становится новым предполагаемым минимумом.

- **Вычисление суммы элементов массива**

Сначала переменной, выделенной под хранение суммы, присваивается значение 0. Затем, при последовательном переборе элементов массива, значение суммы складывается со значением текущего элемента массива и становится новым значением суммы.

- **Вычисление произведения элементов массива**

Сначала переменной, выделенной под хранение произведения, присваивается значение 1. Затем, при последовательном переборе элементов массива, значение произведения умножается на значение текущего элемента массива и становится новым значением произведения.

- **Вычисление среднего значения**

При вычислении среднего значения выполняется вычисление суммы, а затем сумма делится на количество элементов массива.

Необходимо учитывать, что если в используемом языке программирования есть готовые функции для обработки массивов, то их использовать при решении задачи нельзя.

Пример 8.33

Дан целочисленный массив из 20 элементов. Элементы массива могут принимать целые значения от $-10\,000$ до $10\,000$ включительно. Опишите на естественном языке или на одном из языков программирования алгоритм, позволяющий найти и вывести количество пар элементов массива, в которых хотя бы одно число делится на 3. В данной задаче под парой подразумевается два подряд идущих элемента массива. Например, для массива из пяти элементов: 6; 2; 9; -3; 6 — ответ: 4.

Исходные данные объявлены так, как показано ниже на примерах для некоторых языков программирования и естественного языка. Запрещается использовать переменные, не описанные ниже, но разрешается не использовать некоторые из описанных переменных.

Бейсик	Python
<pre> CONST N ASINTEGER = 20 DIM A (1 TO N) ASINTEGER DIM I ASINTEGER, J ASINTEGER, K ASINTEGER FOR I = 1 TO N INPUT A(I) NEXT I ... END </pre>	<pre> # допускается также # использовать две # целочисленные переменные j и k a = [] n = 20 for i in range(0, n): a.append(int(input())) ... </pre>

Паскаль	Алгоритмический язык
<pre> const N = 20; var a: array [1..N] of integer; i, j, k: integer; begin for i := 1 to N do readln(a[i]); ... end.</pre>	<pre> алг нач цел N = 20 целтаб a[1:N] цел i, j, k нцдля i от 1 до N ввод a[i] кц ... кон</pre>
Си	Естественный язык
<pre> #include #define N 20 int main() { int a[N]; int i, j, k; for (i = 0; i < N; i++) scanf("%d", &a[i]); ... return 0; }</pre>	Объявляем массив А из 20 элементов. Объявляем целочисленные переменные I, J, K. В цикле от 1 до 20 вводим элементы массива А с 1-го по 20-й. ...

В качестве ответа вам необходимо привести фрагмент программы (или описание алгоритма на естественном языке), который должен находиться на месте многоточия. Вы можете записать решение также на другом языке программирования (укажите название и используемую версию языка программирования, например *Free Pascal 2.6*) или в виде блок-схемы. В этом случае вы должны использовать те же самые исходные данные и переменные, какие были предложены в условии (например, в образце, записанном на естественном языке).

*Решение***Пример программы на языке Паскаль**

```

k := 0;
for i := 1 to N-1 do
    if (a[i] mod 3=0) or (a[i+1] mod 3=0) then
        inc(k);
writeln(k);

```

Пример 8.34

Дан целочисленный массив из 30 элементов. Элементы массива могут принимать целые значения от 0 до 1000. Элемент массива называется хорошим, если это двузначное число, причем цифра в разряде десятков больше, чем цифра в разряде единиц. Опишите на одном из языков программирования алгоритм, позволяющий найти и вывести сумму всех хороших элементов массива.

Исходные данные объявлены так, как показано ниже. Запрещается использовать переменные, не описанные ниже, но разрешается не использовать часть из описанных переменных.

Бэйсик	Паскаль
<pre> N=30 DIM A(N) AS INTEGER DIM I, J, S AS INTEGER FOR I = 1 TO N INPUT A(I) NEXT I ... END </pre>	<pre> const N=30; var a: array [1..N] of integer; i, j, s: integer; begin for i:=1 to N do readln(a[i]); end. </pre>

Си	Алгоритмический язык
<pre>#include <stdio.h> #define N 30 void main(){ int a[N]; int i, j, s; for (i=0; i < N; i++) scanf("%d", &a[i]); ...</pre>	<pre>алг нач цел N=30 целтаб a[1:N] цел i, j, s нцдля i от 1 до N ввод a[i] кц кон</pre>

В качестве ответа вам необходимо привести фрагмент программы, который должен находиться на месте многоточия. Вы можете записать решение также на другом языке программирования (укажите название и используемую версию языка программирования, например, *Free Pascal 2.4*) или в виде блок-схемы. В этом случае вы должны использовать те же самые исходные данные и переменные, какие были предложены в условии.

Решение

Пример программы на языке Паскаль

```
s:=0;
for i:=1 to N do begin
if (10<=a[i]) and (a[i] <= 99) and
((a[i] div 10) > (a[i] mod 10) ) then s:=s+a[i];
end;
writeln(s);
```

Задача № 8

Основные конструкции языка программирования

Для решения этой задачи нужно знать основные конструкции языка программирования:

- объявление переменных;
- оператор присваивания;
- оператор вывода;
- операторы цикла.

Кроме этого, пригодятся математические формулы:
 n -го члена арифметической прогрессии

$$a_n = a_1 + d(n - 1),$$

суммы n членов геометрической прогрессии

$$S_n = \sum_{i=1}^n a_i = a_1 + a_2 + \dots + a_n = \frac{a_1 + a_n}{2} \cdot n = \frac{2a_1 + d(n-1)}{2} \cdot n.$$

Пример 9.1

Запишите число, которое будет напечатано в результате выполнения программы. Для вашего удобства программа представлена на пяти языках программирования.

Бейсик	Python
<pre> DIM S, N AS INTEGER S = 47 N = 1 WHILE S > 0 S = S - 9 N = N + 4 WEND PRINT(N) </pre>	<pre> s = 47 n = 1 while s > 0: s = s - 9 n = n + 4 print(n) </pre>

Паскаль	Алгоритмический язык
<pre> var s, n: integer; begin s := 47; n := 1; while s > 0 do begin s := s - 9; n := n + 4 end; writeln(n) end.</pre>	<pre> алг нач цел s, n s := 47 n := 1 нцпока s > 0 s := s - 9 n := n + 4 кц вывод n кон</pre>
Си	
<pre> #include <stdio.h> void main() { int s, n; s = 47; n = 1; while (s > 0) { s = s - 9; n = n + 4; } printf("%d\n", n); }</pre>	

Решение

Цикл **while** выполняется до тех пор, пока истинно условие $s > 0$, т.е. переменная s определяет, сколько раз выполнится цикл. Поскольку изначально $s = 47$, а затем уменьшается на 9, цикл выполнится 6 раз, следовательно, $n = 6 \cdot 4 + 1 = 25$.

Ответ: 25.

Пример 9.2

Запишите число, которое будет напечатано в результате выполнения следующей программы. Для вашего удобства программа представлена на пяти языках программирования.

Бейсик	Python
DIM S, N AS INTEGER S = 0 N = 0 WHILE S < 111 S = S + 8 N = N + 2 WEND PRINT N	s = 0 n = 0 while s < 111: s = s + 8 n = n + 2 print(n)
Паскаль	Алгоритмический язык
var s, n: integer; begin s := 0; n := 0; while s < 111 do begin s := s + 8; n := n + 2 end; writeln(n) end.	алг нач цел n, s n := 0 s := 0 нцпока s < 111 s := s + 8 n := n + 2 кц вывод n кон
Си	
<pre>#include int main() {int s = 0, n = 0; while (s < 111) { s = s + 8; n = n + 2; } printf("%d\n", n); return 0; }</pre>	

Решение

Цикл **while** выполняется до тех пор, пока истинно условие $s < 111$, т.е. переменная s определяет, сколько раз выполнится цикл. Поскольку изначально $s = 0$, а затем увеличивается на 8, цикл выполнится 14 раз, следовательно, $n = 2 \cdot 14 = 28$.

Ответ: 28.

Пример 9.3

Запишите число, которое будет напечатано в результате выполнения программы. Для вашего удобства программа представлена на пяти языках программирования.

Бейсик	Python
<pre> DIM N, S AS INTEGER N = 1 S = 0 WHILE N <= 100 S = S + 30 N = N * 3 WEND PRINT S </pre>	<pre> n = 1 s = 0 while n <= 100: s = s + 30 n = n * 3 print(s) </pre>
Паскаль	Алгоритмический язык
<pre> var n, s: integer; begin n := 1; s := 0; while n <= 100 do begin s := s + 30; n := n * 3; end; write(s) end. </pre>	<pre> алг нач цел n, s n := 1 s := 0 нцпока n <= 100 s := s + 30 n := n * 3 кц вывод s кон </pre>
Си	
<pre> #include int main() { int n, s; n = 1; s = 0; while (n <= 100) { s = s + 30; n = n * 3; } printf("%d", s); return 0; } </pre>	

Решение

Цикл **while** выполняется до тех пор, пока истинно условие $n \leq 100$, т.е. переменная n определяет, сколько раз выполнится цикл. Поскольку изначально $n = 1$ и на каждом шаге n увеличивается в три раза, цикл выполнится пять раз. Следовательно, после выполнения цикла s примет значение $30 \cdot 5 = 150$.

Ответ: 150.

Пример 9.4

Запишите число, которое будет напечатано в результате выполнения программы. Для вашего удобства программа представлена на пяти языках программирования.

Бейсик	Python
<pre> DIM S, N AS INTEGER S = 301 N = 0 WHILE S > 0 S = S - 10 N = N + 2 WEND PRINT N </pre>	<pre> s = 301 n = 0 while s > 0: s = s - 10 n = n + 2 print(n) </pre>
Паскаль	Алгоритмический язык
<pre> var s, n: integer; begin s := 301; n := 0; while s > 0 do begin s := s - 10; n := n + 2; end; writeln(n) end. </pre>	<pre> алг нач цел n, s s := 301 n := 0 нцпока s > 0 s := s - 10 n := n + 2 кц вывод n кон </pre>

Си

```
#include <stdio.h>
int main()
{
    int s = 301, n = 0;
    while (s > 0) {
        s = s - 10;
        n = n + 2;
    }
    printf("%d\n", n);
    return 0;
}
```

Решение

Заметим, что после 30 итераций цикла $s = 1$, а $n = 60$. После 31 итерации станет $s = -9$, $n = 62$, и цикл прекратится.

Ответ: 62.

Пример 9.5

Запишите число, которое будет напечатано в результате выполнения программы. Для вашего удобства программа представлена на пяти языках программирования.

Бейсик	Python
DIM S, N AS INTEGER $S = 78$ $N = 0$ WHILE $N < 12$ $S = S - 8$ $N = N + 2$ WEND PRINT(S)	$s = 78$ $n = 0$ while $n < 12$: $s = s - 8$ $n = n + 2$ print(s)

Паскаль	Алгоритмический язык
<pre> var s, n: integer; begin s := 78; n := 0; while n < 12 do begin s := s - 8; n := n + 2 end; writeln(s) end.</pre>	<pre> алг нач цел s, n s := 78 n := 0 нцпока n < 12 s := s — 8 n := n + 2 кц вывод s кон</pre>
Си	
<pre> #include <stdio.h> int main(void) { int s, n; s = 78; n = 0; while (n < 12) { s = s - 8; n = n + 2; } printf("%d\n", s); }</pre>	

Решение

Цикл завершится, когда n станет 12. То есть цикл выполнится 6 раз. $s = 78 - 8 \cdot 6 = 30$.

Ответ: 30.

Пример 9.6

Определите, что будет напечатано в результате выполнения программы, записанной ниже на разных языках программирования.

Бейсик	Python
<pre> DIM N, S AS INTEGER N = 1 S = 0 WHILE N <= 100 S = S + 30 N = N * 2 WEND PRINT S </pre>	<pre> n = 1 s = 0 while n <= 100: s = s + 30 n = n * 2 print(s) </pre>
Паскаль	Алгоритмический язык
<pre> var n, s: integer; begin n := 1; s := 0; while n <= 100 do begin s := s + 30; n := n * 2 end; write(s) end. </pre>	<pre> алг нач цел n, s n := 1 s := 0 нцпока n <= 100 s := s + 30 n := n * 2 кц вывод s кон </pre>
Сп	
<pre> #include <stdio.h> void main() { int n, s; n = 1; s = 0; while (n <= 100) { s = s + 30; n = n * 2; } printf("%d", s); } </pre>	

Решение

Цикл **while** выполняется до тех пор, пока истинно условие $n \leq 100$, т.е. переменная n определяет, сколько раз вы-

полнится цикл. Поскольку изначально $n = 1$, а затем увеличивается вдвое, цикл выполнится 7 раз, следовательно, $s = 30 \cdot 7 = 210$.

Ответ: 210.

Задача № 21

Анализ программ с подпрограммами

Подпрограмма — это поименованная часть программы, которая может быть многократно вызвана из разных частей программы для выполнения некоторых часто используемых действий.

Подпрограммы бывают двух видов: процедуры и функции.

Процедура имеет ту же структуру, что и основная программа. Отличительной чертой **функции** является то, что она обязательно возвращает одно значение, которое является значением самой этой функции. При этом может принимать на вход любое количество аргументов.

В Паскале функция располагается выше основной программы и оформляется следующим образом (вместо многоточия могут быть любые операторы):

```
function F(x: integer):integer;
begin
  ...
  F:= <результат функции>
end;
```

В заголовке функции записывают имя функции, в скобках — список параметров, далее через двоеточие — тип возвращаемого значения; в приведенном примере функция **F** принимает один целый параметр x , и возвращает целое число.



Результат функции записывается в переменную, имя которой совпадает с именем функции. Объявлять эту переменную не нужно.

Поиск наименьшего значения функции:

Подпрограмма поиска наименьшего значения функции $F(x)$ на интервале $[a, b]$, выглядит следующим образом:

```
M:=a; R:=F(a);  
for t:=a to b do  
  if F(t) < R then begin  
    R:=F(t); M:=t;  
  end;
```

Поиск наибольшего значения функции:

Подпрограмма поиска наибольшего значения функции $F(x)$ на интервале $[a, b]$, выглядит следующим образом:

```
M:=a; R:=F(a);  
for t:=a to b do  
  if F(t) > R then begin  
    R:=F(t); M:=t;  
  end;
```

Поиск наибольшего значения функции

Пример 9.7

Определите, какое число будет напечатано в результате выполнения следующего алгоритма (для вашего удобства алгоритм представлен на четырех языках):

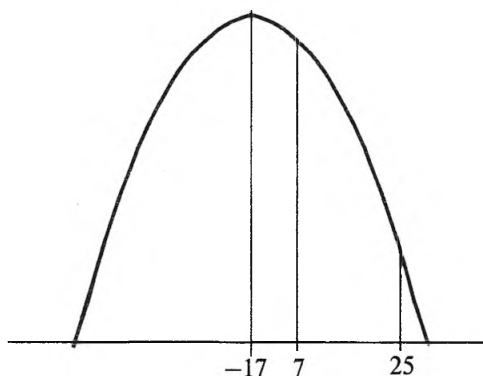
Бейсик	Паскаль
<pre> DIM A, B, T, M, R AS INTEGER A = -7: B = 25 M = A: R = F(A) FOR T = A TO B IF F(T) > R THEN M = T R = F(T) ENDIF NEXT T PRINT R FUNCTION F(x) F = 281 - 2 * (17 + x) * (17 + x); ENDFUNCTION </pre>	<pre> var a,b,t,M,R : integer; Function F(x:integer):integer; begin F := 281 - 2 * (17 + x) * (17 + x); end; begin a := -7; b := 25; M := a; R := F(a); for t := a to b do begin if (F(t) > R) then begin M := t; R := F(t) end end; write(R); end. </pre>
Си	Алгоритмический язык
<pre> #include<stdio.h> int F(int x) { return 281 - 2 * (17 + x) * (17 + x); } void main() { int a, b, t, M, R; a = -7; b = 25; M = a; R = F(a); for (t = a; t <= b; t++){ if (F(t) > R) { M = t; R = F(t); } } printf("%d", R); } </pre>	<pre> алг нач цел a, b, t, R, M a := -7; b := 25 M := a; R := F(a) нц для t от a до b если F(t) > R то M := t; R := F(t) все кц вывод R кон алг цел F(цел x) нач знач := 281 - 2 * (17 + x) * (17 + x) кон </pre>

Решение

Алгоритм предназначен для поиска наибольшего значения функции $F(x)$ на отрезке от a до b .

$$F := 281 - 2 * (17 + x) * (17 + x);$$

$F(x)$ — квадратичная функция с отрицательным старшим коэффициентом. Ее график — парабола с ветвями вниз и вершиной в точке -17 (рисунок выполнен без соблюдения пропорций).



От -17 до бесконечности функция убывает, значит на отрезке $[-7; 25]$ наибольшее значение достигается в левой границе интервала и равно $F(-7) = 81$.

Ответ: 81.

Пример 9.8

Напишите в ответе число, которое будет напечатано в результате выполнения следующего алгоритма (для вашего удобства алгоритм представлен на четырех языках).

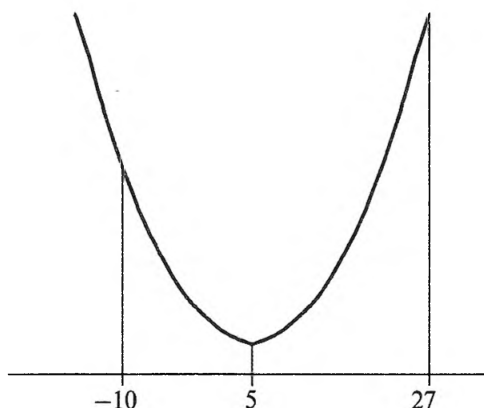
Бейсик	Паскаль
<pre> DIM A, B, T, M, R AS INTEGER A = -10: B = 27 M = A: R = F(A) FOR T = A TO B IF F(T) > R THEN M = T R = F(T) ENDIF NEXT T PRINT M FUNCTION F(x) F = 2*(x - 5)*(x - 5) + 55 ENDFUNCTION </pre>	<pre> var a,b,t,M,R :integer; Function F(x:integer):integer; begin F := 2*(x - 5)*(x - 5) + 55 end; begin a := -10; b := 27; M := a; R := F(a); for t := a to b do begin if (F(t) > R) then begin M := t; R := F(t) end end; write(M) end. </pre>
Сн	Алгоритмический
<pre> #include <stdio.h> int F(int x) { return 2*(x - 5)*(x - 5) + 55; } void main() { int a, b, t, M, R; a = -10; b = 27; M = a; R = F(a); for (t = a; t <= b; t++) { if (F(t) > R) { M = t; R = F(t); } } printf("%d", M); } </pre>	<pre> алг нач цел a, b, t, M, R a := -10; b := 27 M := a; R := F(a) нц для t от a до b если F(t) > R то M := t; R := F(t) все кц вывод M кон алг цел F(цел x) нач знач := 2*(x - 5)*(x - 5) + 55 кон </pre>

Решение

Алгоритм ищет значение аргумента t , при котором функция $F(t)$ имеет наибольшее значение на отрезке от a до b . Если наибольшее значение будет достигнуто в нескольких точках, то выведется первая (левая), так как стоит строгий знак $>$.

$$F := 2*(x - 5)*(x - 5) + 55$$

Функция $F(x)$ — квадратичная функция с положительным старшим коэффициентом. Ее график — парабола с вершиной в точке $x = 5$ и ветвями, направленными вверх.



Расстояние от вершины параболы до правой границы отрезка (27) больше, чем до левой границы (−10), значит, значение в точке 27 наибольшее на отрезке (рисунок выполнен без соблюдения пропорций).

Таким образом, программа выведет на экран число 27.

Ответ: 27.

Пример 9.9

Напишите в ответе число, которое будет напечатано в результате выполнения следующего алгоритма (для вашего удобства алгоритм представлен на четырех языках).

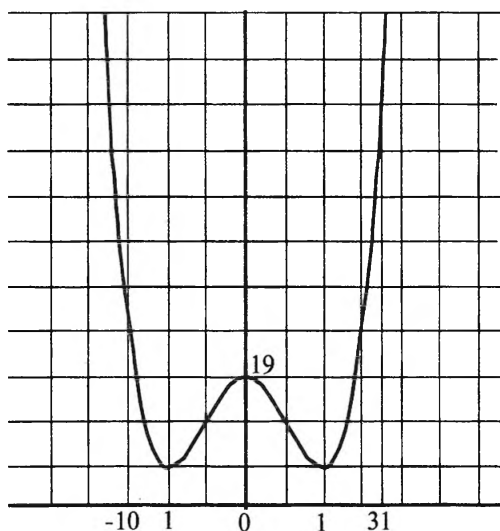
Бейсик	Паскаль
<pre> DIM A, B, T, M, R AS INTEGER A = -10: B = 31 M = A: R = F(A) FOR T = A TO B IF F(T) > R THEN M = T R = F(T) ENDIF NEXT T PRINT M FUNCTION F(x) F = 2*(x*x-1)*(x*x-1)+17 ENDFUNCTION </pre>	<pre> var a,b,t,M,R :integer; Function F(x:integer):integer; begin F := 2*(x*x-1)*(x*x-1)+17 end; begin a := -10; b := 31; M := a; R := F(a); for t := a to b do begin if F(t) > R then begin M := t; R := F(t) end end; write(M) end. </pre>
Си	Алгоритмический язык
<pre> #include <stdio.h> int F(int x) { return 2*(x*x-1)*(x*x-1)+17; } void main() { int a, b, t, M, R; a = -10; b = 31; M = a; R = F(a); for (t = a; t <= b; t++) { if (F(t) > R) { M = t; R = F(t); } } printf("%d", M); } </pre>	<pre> алг нач цел а, b, t, M, R а := -10; b := 31 M := а; R := F(a) нц для t от а до b если F(t) > R то M := t; R := F(t) все кц вывод M кон алг цел F(цел x) нач знач := 2*(x*x-1)*(x*x-1)+17 кон </pre>

Решение

Алгоритм ищет значение аргумента t , при котором функция $F(t)$ имеет наибольшее значение на отрезке от a до b . Если наибольшее значение будет достигнуто в нескольких точках, то выведется первая (левая), так как стоит строгий знак $>$.

$$F := 2*(x*x-1)*(x*x-1)+17$$

Функция $F(x)$ — биквадратная функция с точками минимума -1 и 1 , точкой максимума 0 (рисунок выполнен без соблюдения пропорций).



Точка 31 удалена от начала координат на большее значение, чем точка -10 , значит значение функции в ней наибольшее.

Ответ: 31.

Поиск наименьшего значения функции

Пример 9.10

Определите, какое число будет напечатано в результате выполнения следующего алгоритма (для вашего удобства алгоритм представлен на четырех языках).

Бейсик	Паскаль
<pre> DIM A, B, T, M, R AS INTEGER A = -20: B = 20 M = A: R = F(A) FOR T = A TO B IF F(T) < R THEN M = T R = F(T) ENDIF NEXT T PRINT M FUNCTION F(x) F = 3*(x-8)*(x-8) ENDFUNCTION </pre>	<pre> var a,b,t,M,R :integer; Function F(x:integer):integer; begin F := 3*(x-8)*(x-8) end; begin a := -20; b := 20; M := a; R := F(a); for t := a to b do begin if (F(t) < R) then begin M := t; R := F(t) ;end end; write(M); end. </pre>
Си	Алгоритмический язык
<pre> #include<stdio.h> int F(int x) { return 3*(x-8)*(x-8); } void main() { int a, b, t, M, R; a = -20; b = 20; M = a; R = F(a); for (t=a; t<=b; t++){ if (F(t) < R) { M = t; R = F(t); } } printf("%d", M); } </pre>	<pre> алг нач цел a, b, t, R, M a := -20; b := 20 M := a; R := F(a) нц для t от a до b если F(t) < R то M := t; R := F(t) все кц вывод M кон алг цел F(цел x) нач знач := 3*(x-8)*(x-8) кон </pre>

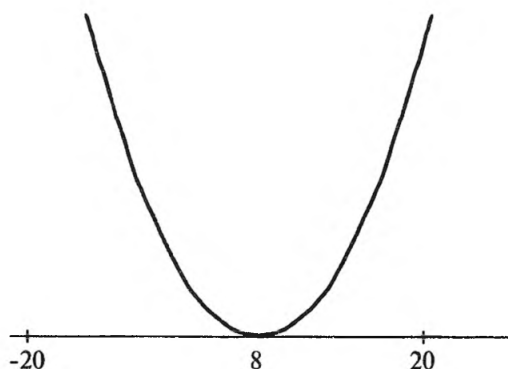
Решение

Алгоритм ищет значение аргумента t , при котором функция $F(t)$ имеет наименьшее значение на отрезке от a до b .

Если наименьшее значение будет достигнуто в нескольких точках, то выведется первая (левая), так как стоит строгий знак $<$.

$$F := 3 \cdot (x-8) \cdot (x-8)$$

Функция $F(x)$ — квадратичная функция с положительным старшим коэффициентом. Ее график — парабола с вершиной в точке $x = 8$ и ветвями, направленными вверх (рисунок выполнен без соблюдения пропорций).



Наименьшее на интервале значение функции $F(x)$ достигается в вершине $x = 8$.

Ответ: 8.

Пример 9.11

Напишите в ответе число, которое будет напечатано в результате выполнения следующего алгоритма (для вашего удобства алгоритм представлен на четырех языках).

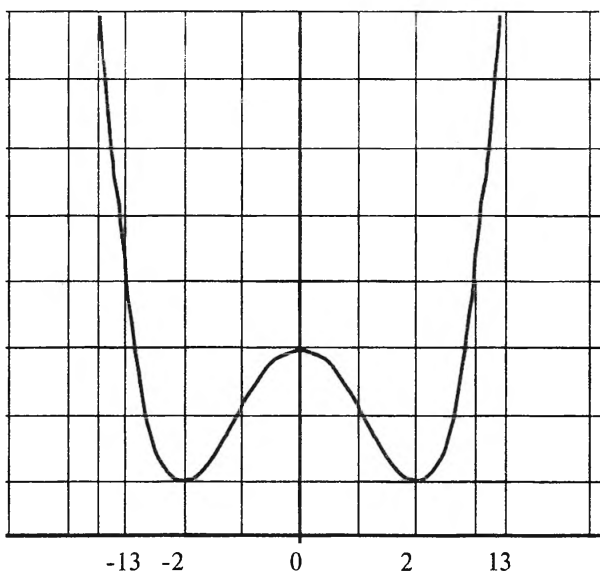
Бейсик	Паскаль
<pre> DIMA, B, T, M, R ASINTEGER A = -13: B = 13 M = A: R=F(A) FOR T = A TO B IFF(T) <RTHEN M = T R = F (T) ENDIF NEXTT PRINT M+18 FUNCTION F(x) F = (x*x-4)*(x*x-4)+11 ENDFUNCTION </pre>	<pre> var a,b,t,M,R :integer; Function F(x:integer):integer; begin F := (x*x-4)*(x*x-4)+11 end; begin a := -13; b := 13; M := a; R := F(a); for t := a to b do begin if (F(t) < R) then begin M := t; R := F(t) end end; write(M+18) end. </pre>
Сн	Алгоритмический
<pre> #include <stdio.h> int F(int x) { return (x*x-4)*(x*x-4)+11; } void main() { int a, b, t, M, R; a = -13; b = 13; M = a; R = F (a); for (t = a; t <= b; t++) { if (F(t) < R) { M = t; R = F(t); } } printf("%d", M+18); } </pre>	<pre> алг нач цел a, b, t, M, R a := -13; b := 13 M := a; R := F(a) нц для t от a до b если F(t) < R то M := t; R := F(t) все кц вывод M+18 кон алг цел P(цел x) нач знач := (x*x-4)* (x*x-4)+11 кон </pre>

Решение

Алгоритм ищет значение аргумента t , при котором функция $F(t)$ имеет наибольшее значение на отрезке от a до b . Если наибольшее значение будет достигнуто в нескольких точках, то выведется первая (левая), так как стоит строгий знак $>$.

$$F := (x^2 - 4)(x^2 - 4) + 11$$

Функция $F(x)$ — биквадратная функция с точками минимума -2 и 2 , точкой максимума 0 (рисунок выполнен без соблюдения пропорций).



На отрезке $[-13 ; 13]$ первый раз наименьшее значение функции достигается в точке -2 . То есть $M = -2$. На печать выводится $M + 18 = 16$.

Ответ: 16.

Пример 9.12

Определите, какое число будет напечатано в результате выполнения следующего алгоритма (для вашего

удобства алгоритм представлен на четырех языках программирования).

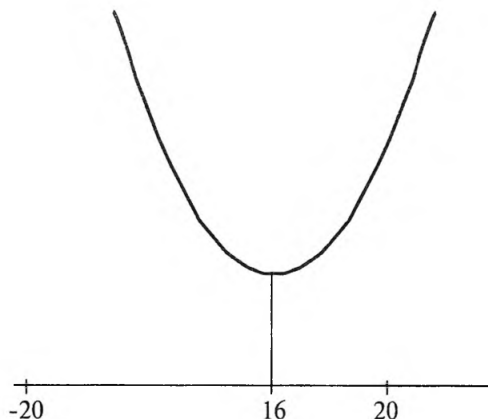
Бейсик	Паскаль
<pre> DIM A, B, T, M, R ASINTEGER A = -20: B = 20 M = A: R = F(A) FOR T = A TO B IF F(T) < R THEN M = T R = F(T) ENDIF NEXT T PRINT M FUNCTION F(x) F = 2*(x-15)*(x-17)+19 ENDFUNCTION </pre>	<pre> var a,b,t,M,R :integer; Function F(x:integer):integer; begin F := 2*(x-15)*(x-17)+19 end; BEGIN a := -20; b := 20; M := a; R := F(a); for t := a to b dobegin if (F(t) < R) thenbegin M := t; R := F(t) end end; write(M) END. </pre>
Си	Алгоритмический язык
<pre> #include <stdio.h> int F(int x) { return 2*(x-15)*(x-17)+19; } void main() { int a, b, t, M, R; a = -20; b = 20; M = a; R = F(a); for (t = a; t <= b; t++) { if (F(t) < R) { M = t; R = F(t); } } printf("%d", M); } </pre>	<pre> алг нач цел a, b, t, R, M a := -20; b := 20 M := a; R := F(a) нцдля t от a до b если F(t) < R то M := t; R := F(t) все кц вывод M кон алгцел F(цел x) нач знач := 2*(x-15)*(x-17)+19 кон </pre>

Решение

Алгоритм ищет значение аргумента t , при котором функция $F(t)$ имеет наименьшее значение на отрезке от a до b . Если наименьшее значение будет достигнуто в нескольких точках, то выведется первая (левая), т.к. стоит строгий знак $<$.

$$F := 2*(x-15)*(x-17)+19 = 2*x^2 - 64*x + 529$$

Функция $F(x)$ - квадратичная функция с положительным старшим коэффициентом. Ее график — парабола с вершиной в точке $x = 64/4 = 16$ и ветвями, направленными вверх (рисунок выполнен без соблюдения пропорций).



На отрезке $[-20 ; 20]$ минимальное значение функции достигается в вершине параболы. То есть в точке 16.

Ответ: 16.

**Поиск значений, при которых программа
выдает тот же результат**

Пример 9.13

Напишите в ответе наименьшее значение входной переменной k , при котором программа выдает тот же ответ, что и при входном значении $k = 10$. Для вашего

удобства программа приведена на пяти языках программирования.

Бейсик	Python
<pre> DIM K, I AS LONG INPUT K I = 1 WHILE F(I) < G(K) I = I + 1 WEND PRINT I FUNCTION F(N) F = N * N * N ENDFUNCTION FUNCTION G(N) G = 2*N + 3 ENDFUNCTION </pre>	<pre> def f(n): return n*n*n def g(n): return 2*n+3 k = int(input()) i = 1 while f(i) < g(k): i+=1 print (i) </pre>
Паскаль	Алгоритмический язык
<pre> var k, i : longint; function f(n: longint): longint; begin f := n * n * n; end; function g(n: longint): longint; begin g := 2*n + 3; end; begin readln(k); i := 1; while f(i) < g(k) do i := i+1; writeln(i) end. </pre>	<pre> алг нач цел i, k ввод k i := 1 нцпока f(i) < g(k) i := i + 1 кц вывод i кон алгцел f(цел n) нач знач := n * n * n кон алгцел g(цел n) нач знач := 2*n + 3 кон </pre>

```

#include
long f(long n) {
    return n * n * n;
}
long g(long n) {
    return 2*n + 3;
}
int main()
{
    long k, i;
    scanf("%ld", &k);
    i = 1;
    while(f(i) < g(k))
        i++;
    printf("%ld", i);
    return 0;
}

```

Решение

Программа сравнивает значения функций $f(i) = i^3$ и $g(k) = 2 \cdot k + 3$, увеличивает i на единицу, если $i^3 < 2 \cdot k + 3$ и выводит первое значение переменной i при котором

$$i^3 \geq 2 \cdot k + 3.$$

При $k = 10$ программа выведет число 3:

$$2^3 \leq 2 \cdot k + 3 < 3^3.$$

Отсюда наименьшее $k = 3$.

Ответ: 3.

Пример 9.14

Напишите в ответе число различных значений входной переменной k , при которых программа выдает тот же ответ, что и при входном значении $k = 64$. Значение $k = 64$ также включается в подсчет различных значений k . Для вашего удобства программа приведена на пяти языках программирования.

Бейсик	Python
<pre> DIM K, I AS LONG INPUT K I = 12 WHILE I > 0 AND F(I) >= K I = I - 1 WEND PRINT I FUNCTION F(N) F = N * N ENDFUNCTION </pre>	<pre> def f(n): return n * n k = int(input()) i = 12 while i > 0 and f(i) >= k: i = i - 1 print(i) </pre>
Паскаль	Алгоритмический язык
<pre> var k, i : longint; function f(n: longint) : longint; begin f := n * n end; begin readln(k); i := 12; while (i>0) and (f(i)>=k) do i := i-1; writeln(i) end. </pre>	<pre> нач цел i, k ввод k i := 12 нцпока i > 0 и f(i) >= k i := i - 1 кц вывод i кон алгцел f(цел n) нач знач := n * n кон </pre>
Си	
<pre> #include <stdio.h> int f(int n) { return n * n; } int main(void) { int k, i; scanf("%d", &k); i = 12; while (i > 0 && f(i) >= k) i--; printf("%d", i); } </pre>	

Решение:

При $k = 64 = 8^2$, программа выведет число 7. Переменная k отвечает за количество выполнений цикла.

Заметим, что при любом k , большем чем 64, программа выведет число больше 7, а при $k = 7^2$ — меньше. Следовательно, нам подходят все целые числа, меньшие или равные 64, но большие 49. Таких чисел 15.

Ответ: 15.

Пример 9.15

При каком наибольшем значении входной переменной k программа выдает тот же ответ, что и при входном значении $k = 64$? Для вашего удобства программа приведена на пяти языках программирования.

Бейсик	Python
<pre> DIM K, I AS LONG INPUT K I = 12 WHILE I > 0 AND F(I) >= K I = I - 1 WEND PRINT I FUNCTION F(N) F = N * N - 20 ENDFUNCTION </pre>	<pre> def f(n): return n * n - 20 k = int(input()) i = 12 while i > 0 and f(i) >= k: i = i - 1 print(i) </pre>
Алгоритмический язык	Паскаль
<pre> алг нач цел i, k ввод k i := 12 нцпока i > 0 и f(i) >= k i := i - 1 кц вывод i кон </pre>	<pre> var k, i : longint; function f(n: longint) : longint; begin f := n * n - 20 end; begin readln(k); i := 12; </pre>

алгцел f(цел n) нач знач := n * n - 20 кон	while (i>0) and (f(i) >= k) do i := i-1; writeln(i) end.
C++	
<pre>#include <stdio.h> long f(long n) {return n * n - 20; } int main() { long k, i; scanf("%ld", &k); i = 12; while (i > 0 && f(i) >= k) i--; printf("%ld", i); return 0; }</pre>	

Решение

Программа выводит максимальное значение переменной i , не большее 12, которое удовлетворяет неравенству $i^2 - 20 < k$.

Для $k = 64$ $i = 9$;

$9^2 - 20 < k \leq 10^2 - 20$

$61 < k \leq 80$.

То есть такой же ответ ($i = 9$), как при $k = 64$, программа выведет для $62 \leq k \leq 80$.

Значит, наибольшее $k = 80$.

Ответ: 80.

Задача № 24

Поиск и исправление ошибок в программе

Самая распространенная ошибка, которую нужно найти и исправить, — это неправильное использование вложенных условных операторов. Для усложнения поиска возмож-

но неправильное форматирование текста (неправильно поставленные отступы).

Для того чтобы найти ошибку, нужно поставить в соответствие друг другу все части условного оператора **if** и **else**.

Помним, что часть **else** относится к ближайшему **if**. При этом наличие части **else** не обязательно.

Кроме того, часто присутствует ошибка при вводе или выводе. Обязательно нужно проверить, та ли информация выводится на экран.

Особого внимания требует инициализация переменных.

Формат книги не позволяет рассмотреть все основные типы задач 2 части, рассмотрим лишь те, которые встречались на проверочных и экзаменационных работах последних двух лет.

Пример 9.16

На обработку поступает положительное целое число, не превышающее 10^9 . Нужно написать программу, которая выводит на экран сумму цифр этого числа, меньших 7. Если в числе нет цифр, меньших 7, требуется на экран вывести 0. Программист написал программу неправильно. Ниже эта программа для вашего удобства приведена на пяти языках программирования.

Бейсик	Python
<pre> DIM N, DIGIT, SUM AS LONG INPUT N SUM = 0 WHILE N > 0 DIGIT = N MOD 10 IF DIGIT < 7 THEN SUM = SUM + 1 ENDIF N = N \ 10 WEND PRINT DIGIT </pre>	<pre> N = int(input()) sum = 0 while N > 0: digit = N % 10 if digit < 7: sum = sum + 1 N = N // 10 print(digit) </pre>

Паскаль	Алгоритмический язык
<pre> var N, digit, sum: longint; begin readln(N); sum := 0; while N > 0 do begin digit := N mod 10; if digit < 7 then sum := sum + 1; N := N div 10; end; writeln(digit) end. </pre>	<pre> алг нач цел N, digit, sum ввод N sum := 0 нцпока N > 0 digit := mod(N,10) если digit < 7 то sum := sum + 1 все N := div(N,10) кц вывод digit кон </pre>
C++	
<pre> #include int main() { int N, digit, sum; scanf("%d", &N); sum = 0; while (N > 0) { digit = N % 10; if (digit < 7) sum = sum + 1; N = N / 10; } printf("%d",digit); return 0; } </pre>	

Последовательно выполните следующее.

1. Напишите, что выведет эта программа при вводе числа 456.
2. Приведите пример такого трехзначного числа, при вводе которого программа выдает верный ответ.

3. Найдите все ошибки в этой программе (их может быть одна или несколько). Известно, что каждая ошибка затрагивает только одну строку и может быть исправлена без изменения других строк. Для каждой ошибки:

- 1) выпишите строку, в которой сделана ошибка;
- 2) укажите, как исправить ошибку, т.е. приведите правильный вариант строки.

Достаточно указать ошибки и способ их исправления для одного языка программирования. Обратите внимание, что требуется найти ошибки в имеющейся программе, а не написать свою, возможно, использующую другой алгоритм решения. Исправление ошибки должно затрагивать только строку, в которой находится ошибка.

Решение

Решение использует запись программы на Паскале. Допускается использование программы на любом из четырех других языков.

1. Программа выведет число 4.
2. Пример числа, при вводе которого программа выдает верный ответ: 835.

Программа работает неправильно из-за неверной выводимой на экран переменной и неверного увеличения суммы. Соответственно, программа будет работать верно, если в числе старшая цифра (крайняя левая) равна сумме цифр, меньших 7.

3. В программе есть две ошибки.

Первая ошибка. Неверное увеличение суммы.

Строка с ошибкой:

```
sum := sum + 1;
```

Верное исправление:

```
sum := sum + digit;
```

Вторая ошибка. Неверный вывод ответа на экран.

Строка с ошибкой:

```
writeln(digit)
```

Верное исправление:

```
writeln(sum)
```

Пример 9.17

Для заданного положительного вещественного числа A необходимо найти максимальное целое число K , при котором выполняется неравенство

$$1 + \frac{1}{2} + \frac{1}{3} + \dots + \frac{1}{K} < A$$

(при $K = 0$ сумма считается равной 0).

Для решения этой задачи ученик написал такую программу.

Бейсик	Python
<pre> DIM A, S AS DOUBLE DIM K AS INTEGER INPUT A K = 1 S = 1 WHILE S < A S = S + 1.0/K K = K + 1 WEND PRINT K END </pre>	<pre> a = float(input()) k = 1 s = 1 while s < a: s = s + 1.0/k k = k + 1 print(k) </pre>
Алгоритмический язык	Паскаль
<pre> алг нач вещ a, s цел k ввод a k := 1 s := 1 </pre>	<pre> var a, s: real; k: integer; begin read(a); k := 1; s := 1; </pre>

начало $s < a$ $s := s + 1.0/k$ $k := k + 1$ кц вывод k кон	while $s < a$ do $s := s + 1.0/k$; $k := k + 1$; end ; write (k); end .
Си	
<pre>#include <stdio.h> int main(){ double a, s; int k; scanf("%lf",&a); k = 1; s = 1; while (s<a) { s = s + 1.0/k; k = k + 1; } printf("%d", k); return 0; }</pre>	

Последовательно выполните следующее.

1. Напишите, что выведет эта программа при вводе числа 1.2.
2. Приведите пример числа, при вводе которого программа даст верный ответ.
3. Найдите в программе все ошибки (их может быть одна или несколько).

Для каждой ошибки выпишите строку, в которой она допущена, и приведите эту же строку в исправленном виде.

Обратите внимание: вам нужно исправить приведенную программу, а не написать свою. Вы можете только исправлять ошибочные строки; удалять строки или добавлять новые строки нельзя. Постарайтесь также не внести новые ошибки — за это оценка снижается.

Решение

Решение использует запись программы на Паскале. Допускается использование программы на других языках.

1. При вводе числа 1.2 программа выведет число 2.

2. Примеры чисел, при вводе которых программа выводит верный ответ: 1.6, 2.05.

Программа содержит две ошибки, одна из которых приводит к увеличению ответа, другая — к уменьшению.

В некоторых случаях эти ошибки компенсируют друг друга, и ответ оказывается правильным. Это происходит, если значение A попадает в один из следующих диапазонов: $1.5 < A < 1.83$, $2 < A < 2.08$.

3. Программа содержит две ошибки.

1) *Неверная инициализация.* Начальное значение S должно быть равно нулю.

В приведенном варианте вычисленная сумма оказывается на 1 больше правильного значения.

Строка с ошибкой:

`s := 1;`

Правильная строка:

`s := 0;`

2) *Неверное определение ответа.* Приведенная программа находит не максимальное K , при котором выполняется неравенство, а минимальное, при котором оно не выполняется, т.е. увеличивает верное значение на 1.

Кроме того, использованный порядок действий в цикле (увеличение K после увеличения S) приводит к увеличению еще на 1. Это можно было бы исправить, изменив порядок действий в цикле и уменьшив K после завершения цикла, но эти действия не разрешены по условию задачи.

Поэтому для исправления ошибки можно просто скорректировать значение при выводе.

Строка с ошибкой:

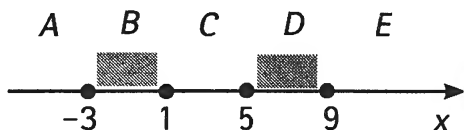
`write(k);`

Правильная строка:

`write(k-2);`

Пример 9.18

Требовалось написать программу, при выполнении которой с клавиатуры считывается координата точки на прямой (x — действительное число) и определяется принадлежность этой точки одному из выделенных отрезков B и D (включая границы). Программист торопился и написал программу неправильно.



Бейсик	Паскаль
<pre> INPUT x IF x>=-3 THEN IF x<=9 THEN IF x>1 THEN PRINT "не принадлежит" ELSE PRINT "принадлежит" ENDIF ENDIF ENDIF END </pre>	<pre> var x: real; begin readln(x); if x>=-3 then if x<=9 then if x>1 then write('не принадлежит') else write('принадлежит') end. end. </pre>
Си	Алгоритмический язык
<pre> void main(void) { float x; scanf("%f",&x); if(x>=-3) if(x<=9) if(x>1) printf("не принадлежит"); else printf("принадлежит"); } </pre>	<pre> алг нач вещ x ввод x если x>=-3 то если x<=9 то если x>1 то вывод 'не принадлежит' иначе вывод 'принадлежит' все все все кон </pre>

Последовательно выполните следующее.

1. Перерисуйте и заполните таблицу, которая показывает, как работает программа при аргументах, принадлежащих различным областям (A , B , C , D и E). Границы (точки -3 , 1 , 5 и 9) принадлежат заштрихованным областям (B и D соответственно).

В столбцах условий укажите «Да», если условие выполнится; «Нет», если условие не выполнится; «—» (прочерк), если условие не будет проверяться; «не изв.», если программа ведет себя по-разному для разных значений, принадлежащих данной области. В столбце «Программа выведет» укажите, что программа выведет на экран. Если программа ничего не выводит, напишите «—» (прочерк). Если для разных значений, принадлежащих области, будут выведены разные тексты, напишите «не изв.». В последнем столбце укажите «Да» или «Нет».

2. Укажите, как нужно доработать программу, чтобы не было случаев ее неправильной работы. (Это можно сделать несколькими способами, достаточно указать любой способ доработки исходной программы.)

Область	Условие 1 ($x \geq -3$)	Условие 2 ($x \leq 9$)	Условие 3 ($x > 1$)	Программа выведет	Область обрабатывается верно
A					
B					
C					
D					
E					

Решение

1.

Об- ласть	Условие 1 ($x \geq -3$)	Условие 2 ($x \leq 9$)	Условие 3 ($x > 1$)	Программа выведет	Область обрабатыва- ется верно
A	нет	–	–	–	нет
B	да	да	нет	принадлежит	да
C	да	да	да	не принадлежит	нет
D	да	да	да	не принадлежит	да
E	да	нет	–	–	нет

2. Возможная доработка (Паскаль):

```
if (x >= -3) and (x <= 1) or (x >= 5) and (x <= 9) then
  write('принадлежит')
else
  write('не принадлежит')
```

Задача № 27

Разработка программы средней сложности

Разработка программы — это самая сложная и наиболее высоко оцениваемая задача экзаменационной работы. Она оценивается следующим образом:

- 4 балла, если написанная программа работает верно, она эффективна и содержит до 3-х синтаксических ошибок;
- 3 балла, если написанная программа работает верно, она неэффективна по памяти (но эффективна по времени), содержит не более 5-ти синтаксических ошибок и не более одной смысловой ошибки;
- 2 балла, если написанная программа работает верно, но она неэффективна, содержит не более 7-ми синтаксических ошибок и не более двух смысловых ошибок;

- 1 балл, если программа не написана или работает неверно, однако алгоритм решения описан правильно.

Давайте выделим красугольные камни этой самой сложной задачи.

- **Правильный алгоритм.**

До того, как начать программировать, нужно хорошо понять, что собственно мы собираемся запрограммировать. Причем продумать алгоритм нужно до мелочей, учесть все возможные варианты поведения программы. После этого обязательно подробно и понятно записать алгоритм на простом языке, в виде блок-схемы или в виде таблицы. Это описание будет путеводной нитью при разработке программы. А заодно принесет 1 балл.

- **Эффективность.**

В постановке задачи требуется не просто написать программу, а написать **эффективную** программу. Давайте разберемся, что же такое эффективность. Эффективность в данном смысле — это умение экономно расходовать основные ресурсы: память компьютера и время.

Наиболее ценным ресурсом в этой задаче считается время. Эффективность по времени расценивается «дороже», чем эффективность по памяти. Для минимизации времени нужно стараться избегать вложенных циклов.

Для экономии памяти следует: правильно выбирать тип переменной, при возможности не сохранять вводимые данные в массив или переменные, а анализировать сразу при вводе, многократно использовать переменные (если возможно, использовать одну переменную для разных целей).

Формат книги не позволяет рассмотреть все основные типы и приемы решения этой задачи. Ограничимся лишь анализом некоторых задач 2014–2015 года.

Пример 9.19

В физической лаборатории проводится долговременный эксперимент по изучению гравитационного поля Земли. По каналу связи каждую минуту в лабораторию передается положительное целое число — текущее показание прибора «Сигма 2015». Количество передаваемых чисел в серии известно и не превышает 10 000. Все числа не превышают 1000. Временем, в течение которого происходит передача, можно пренебречь.

Необходимо вычислить «бета-значение» серии показаний прибора — минимальное четное произведение двух показаний, между моментами передачи которых прошло не менее 6 минут. Если получить такое произведение не удается, ответ считается равным -1 .

Вам предлагается два задания, связанных с этой задачей: задание А и задание Б. Вы можете решать оба задания или одно из них по своему выбору. Итоговая оценка выставляется как максимальная из оценок за задания А и Б. Если решение одного из заданий не представлено, то считается, что оценка за это задание — 0 баллов. Задание Б является усложненным вариантом задания А, оно содержит дополнительные требования к программе.

А. Напишите на любом языке программирования программу для решения поставленной задачи, в которой входные данные будут запоминаться в массиве, после чего будут проверены все возможные пары элементов. Перед программой укажите версию языка программирования.

ОБЯЗАТЕЛЬНО укажите, что программа является решением ЗАДАНИЯ А.

Максимальная оценка за выполнение задания А — 2 балла.

Б. Напишите программу для решения поставленной задачи, которая будет эффективна как по времени, так и по памяти (или хотя бы по одной из этих характеристик).

Программа считается эффективной по времени, если время работы программы пропорционально количеству полученных показаний прибора N , т.е. при увеличении N в k раз время работы программы должно увеличиваться не более чем в k раз.

Программа считается эффективной по памяти, если размер памяти, использованной в программе для хранения данных, не зависит от числа N и не превышает 1 килобайта.

Перед программой укажите версию языка программирования и кратко опишите использованный алгоритм.

ОБЯЗАТЕЛЬНО укажите, что программа является решением ЗАДАНИЯ Б.

Максимальная оценка за правильную программу, эффективную по времени и по памяти, — 4 балла.

Максимальная оценка за правильную программу, эффективную по времени, но неэффективную по памяти, — 3 балла. **НАПОМИНАЕМ!** Не забудьте указать, к какому заданию относится каждая из представленных вами программ.

Входные данные представлены следующим образом. В первой строке задается число N — общее количество показаний прибора. Гарантируется, что $N > 6$. В каждой из следующих N строк задается одно положительное целое число — очередное показание прибора.

Пример входных данных:

11

12

45

5

3

17

23

21

20

19

18

17

Программа должна вывести одно число — описанное в условии произведение либо -1 , если получить такое произведение не удастся.

Пример выходных данных для приведенного выше примера входных данных:

54

Решение

А. Пример правильной программы на языке Паскаль. Программа неэффективна ни по времени, ни по памяти

```
const s = 6; {требуемое расстояние между показаниями}
var
    N: integer;
    a: array[1..10000] of integer; {все показания прибора}
    mp: integer; {минимальное значение произведения}
    i, j: integer;
begin
    readln(N);
    {Ввод значений прибора}
    for i:=1 to N do
        readln(a[i]);
        mp := 1000 * 1000 + 1;
        for i := 1 to N-s do begin
            for j := i+s to N do begin
                if (a[i]*a[j] mod 2 = 0) and
(a[i]*a[j] < mp)
                    then mp := a[i]*a[j]
            end;
        end;
    end;
    if mp = 1000 * 1000 + 1 then mp := -1;
    writeln(mp)
end.
```

Б. Пример правильной программы на языке Паскаль.

Программа использует сдвиги, но эффективна по времени и по памяти

Решение

Чтобы произведение было четным, хотя бы один сомножитель должен быть четным, поэтому при поиске подходящих произведений четные показания прибора можно рассматривать в паре с любыми другими, а нечетные — только с четными.

Для каждого показания с номером k , начиная с $k = 7$, рассмотрим все допустимые по условиям задачи пары, в которых данное показание получено вторым. Минимальное произведение из всех этих пар будет получено, если первым в паре будет взято минимальное подходящее показание среди всех, полученных от начала приема и до показания с номером $k - 6$. Если очередное показание четное, минимальное среди предыдущих может быть любым, если нечетное — только четным.

```
const s = 6; {требуемое расстояние между показаниями}
      amax = 1001; {больше максимально возможного по-
казания}
var
  N: integer;
  a: array[1..s] of integer; {хранение s показаний прибора}
  a_: integer; {ввод очередного показания}
  ma: integer; {минимальное число без s последних}
  me: integer; {минимальное четное число без s
                последних}
  mp: integer; {минимальное значение произведения}
  p: integer;
  i, j: integer;
begin
```



```
readln(N);
{Ввод первых s чисел}
for i:=1 to s do readln(a[i]);
{Ввод остальных значений, поиск
минимального произведения}
ma := amax; me := amax;
mp := amax*amax;
for i := s + 1 to N do begin
    readln(a_);
    f a[1] < ma then ma := a[1];
    if (a[1] mod 2 = 0) and (a[1] < me) then
        me := a[1];
    if a_ mod 2 = 0 then p := a_ * ma
    else if me < amax then p := a_ * me
    else p := amax* amax;
    if (p < mp) then mp := p;
    {сдвигаем элементы вспомогательного
    массива влево}
    for j := 1 to s - 1 do
        a[j] := a[j + 1];
    a[s] := a_
end;
if mp = amax*amax then mp:=-1;
writeln(mp)
end.
```

Пример 9.20

Для заданной последовательности целых чисел необходимо найти максимальную сумму квадратов двух ее элементов, номера которых различаются не менее чем на 10. Значение каждого элемента последовательности не превышает 100. Количество элементов последовательности не превышает 10000.

Вам предлагаются два задания, связанные с этой задачей: задание А и задание Б. Вы можете решать оба задания А и Б или одно из них по своему выбору.

Итоговая оценка выставляется как максимальная из оценок за задания А и Б. Если решение одного из заданий не представлено, то считается, что оценка за это задание составляет 0 баллов.

Задание Б является усложненным вариантом задания А, оно содержит дополнительные требования к программе.

А. Напишите на любом языке программирования программу для решения поставленной задачи, в которой входные данные будут запоминаться в массиве, после чего будут проверены все возможные пары элементов.

Перед программой укажите версию языка программирования. Обязательно укажите, что программа является решением задания А.

Максимальная оценка за выполнение задания А — 2 балла.

Б. Напишите программу для решения поставленной задачи, которая будет эффективна как по времени, так и по памяти (или хотя бы по одной из этих характеристик).

Программа считается эффективной по времени, если время работы программы пропорционально количеству элементов последовательности N , т.е. при увеличении N в k раз время работы программы должно увеличиваться не более чем в k раз.

Программа считается эффективной по памяти, если размер памяти, использованной в программе для хранения данных, не зависит от числа N и не превышает 1 килобайта. Перед программой укажите версию языка программирования и кратко опишите использованный алгоритм.

Обязательно укажите, что программа является решением задания Б.

Максимальная оценка за правильную программу, эффективную по времени и по памяти, — 4 балла.

Максимальная оценка за правильную программу, эффективную по времени, но неэффективную по памяти, — 3 балла.

Решение

А. Пример правильной программы на языке Паскаль, не эффективной ни по времени, ни по памяти:

```
Const d = 10;
var
  N: integer;
  a: array[1..10000] of integer;
  {хранение всех элементов}
  m: integer; {максимальное значение суммы квадратов}
  i, j: integer;
begin
  readln(N);
  {Ввод значений элементов}
  for i:=1 to N do readln(a[i]);
  m := 0;
  for i := 1 to N-d do begin
    for j := i+d to N do begin
      if a[i]*a[i]+a[j]*a[j]>m then
        m := a[i]*a[i]+a[j]*a[j];
    end;
  end;
  writeln(m)
end.
```

Б. Для каждого элемента с номером k (нумерацию начинаем с 1), начиная с $k = 11$, рассмотрим все допустимые по условиям задачи пары, в которых данный элемент является вторым.

Максимальная сумма квадратов элементов этих пар будет получена, если первым в паре будет взят максимальный элемент среди всех, от первого и до элемента с номером $k - 10$. Для получения эффективного по времени решения нужно по мере ввода данных помнить максимальное текущее значение, квадрат каждого вновь введенного значения суммировать с квадратом максимума, найденным на 10 элементов ранее, и выбрать максимальную из всех таких сумм.

Поскольку каждое текущее максимальное значение используется после ввода еще 10 элементов и после этого становится ненужным, достаточно хранить только 10 последних максимумов. Для этого можно использовать буферный массив из 10 элементов и циклически заполнять его по мере ввода данных.

Размер этого массива не зависит от общего количества введенных элементов, поэтому такое решение будет эффективным не только по времени, но и по памяти.

Ниже приводится пример такой программы

Программа 1. Пример правильной и эффективной программы на языке Паскаль:

```
program N_27;
const d = 10;
var
  N: integer;
  a: array[0..d-1] of integer; {буфер}
  {k-е введенное число записываем в ячейку a[k mod d]}
  x: integer;
  mx: integer; {максимальное введенное число}
  {(не считая 10 последних)}
```

```
m: integer; {максимальное значение суммы квадратов}
i: integer;
begin
    readln(N);
    {Ввод первых d чисел}
    for i:=1 to d do
        begin
            readln(x);
            a[i mod d] := x
        end;
    { Ввод остальных элементов, поиск максимальной сум-
мы квадратов}
    mx := 0; m := 0;
    for i := d + 1 to N do
        begin
            readln(x);
            if a[i mod d] > mx then mx := a[i mod d];
            if x*x + mx*mx > m then m := x*x + mx*mx;
            a[i mod d] := x
        end;
    writeln(m)
end.
```

Источники заданий

Демонстрационная версия ЕГЭ по информатике 2013–2016 гг.

МИОО. Тренировочные работы по информатике 2013–2015 гг.

МИОО. Диагностические работы по информатике 2013–2015 гг.

СтатГрад. Тренировочные работы по информатике 2013–2015 гг.

СтатГрад. Диагностические работы по информатике 2013–2015 гг.

СтатГрад. Репетиционные работы по информатике 2013–2015 гг.

ЕГЭ по информатике 2013 г.

ЕГЭ по информатике досрочный 2013–2015 гг.

Поляков К.Ю. <http://kpolyakov.spb.ru/>

ФИПИ Открытый банк заданий ЕГЭ <http://www.fipi.ru/content/otkrytyy-bank-zadaniy-ege>

Содержание

ВВЕДЕНИЕ	3
1. СИСТЕМЫ СЧИСЛЕНИЯ.....	6
Задача № 1. Системы счисления. Кодирование чисел в разных системах	6
Задача № 16. Позиционные системы счисления.....	16
2. МОДЕЛИРОВАНИЕ И КОМПЬЮТЕРНЫЙ ЭКСПЕРИМЕНТ	24
Задача № 3. Анализ информационных моделей	24
Задача № 15. Поиск путей в графе	34
3. ТЕХНОЛОГИИ ПОИСКА И ХРАНЕНИЯ ИНФОРМАЦИИ	37
Задача № 4. Файловая система. Базы данных.....	37
Задача № 17. Запросы для поисковых систем	49
4. ОБРАБОТКА ЧИСЛОВОЙ ИНФОРМАЦИИ	57
Задача № 7. Анализ электронных таблиц и диаграмм.....	57
5. АРХИТЕКТУРА КОМПЬЮТЕРОВ И КОМПЬЮТЕРНЫХ СЕТЕЙ	71
Задача № 12. Организация компьютерных сетей. Адресация в Интернете	71



6. ИНФОРМАЦИЯ И ЕЕ КОДИРОВАНИЕ	79
Задача № 5. Кодирование и декодирование информации.....	79
Задача № 9. Кодирование и передача звуковой и графической информации.....	90
Задача № 10. Методы измерения количества информации.....	103
Задача № 13. Информационный объем сообщения.....	112
 7. ЛОГИКА И АЛГОРИТМЫ	 116
Задача № 2. Построение и анализ таблиц истинности логических выражений.....	116
Задача № 18. Основные понятия и законы математической логики	125
Задача № 23. Построение и преобразование логических выражений.....	138
Задача № 11. Рекурсивные алгоритмы.....	150
Задача № 19. Работа с массивами.....	158
Задача № 26. Построение дерева игры. Поиск выигрышной стратегии.....	169
 8. ЭЛЕМЕНТЫ ТЕОРИИ АЛГОРИТМОВ.....	 180
Задача № 6. Анализ, исполнение и построение линейного алгоритма для исполнителя	180
Задача № 14. Исполнение алгоритма с фиксированным набором команд.....	188
Задача № 20. Анализ алгоритма, содержащего цикл и ветвление	203
Задача № 22. Анализ результата исполнения алгоритма.....	218
Задача № 25. Алгоритмы обработки массивов	229

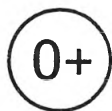


9. ЯЗЫКИ ПРОГРАММИРОВАНИЯ.....	235
Задача № 8. Основные конструкции языка программирования.....	235
Задача № 21. Анализ программ с подпрограммами.....	243
Задача № 24. Поиск и исправление ошибок в программе....	261
Задача № 27. Разработка программы средней сложности ...	270
Источники заданий.....	281

Учебное издание



ЕСАКОВА Лада Борисовна



ИНФОРМАТИКА

АВТОРСКИЙ КУРС ПОДГОТОВКИ К ЕГЭ

Ответственный редактор

А. Яненко

Выпускающий редактор

Г. Логвинова

Технический редактор

Ю. Давыдова

Компьютерная верстка:

А. Патулова

Формат 84x108 ¹/₃₂. Бумага офсетная.

Тираж 2500 экз. Заказ № 18-03-0508.

Импортер на территории ЕАЭС: ООО «Феникс»

344011, Россия, Ростовская обл., г. Ростов-на-Дону, ул. Варфоломеева, 150

Тел./факс: (863) 261-89-50, 261-89-59

Изготовлено в Украине. Дата изготовления: 03.2018

Изготовитель: ООО «БЭТ». 61024, Украина, г. Харьков, ул. Ольминского, 17.