

ЧАСТЬ А

А1. Сколько единиц в двоичной записи десятичного числа 255?

1. 1 2. 2 3. 7 4. 8

Решение: Надо помнить, что $2^k - 1$ это в двоичном представлении **k единиц**. Так как, $255 = 256 - 1 = 2^8 - 1$, то в двоичной записи десятичного числа 255 будет 8 единиц.

Ответ: 4

А2. Между населёнными пунктами А, В, С, D, E, F построены дороги, протяжённость которых приведена в таблице. (Отсутствие числа в таблице означает, что прямой дороги между пунктами нет.)

	А	В	С	D	E	F
А		3				
В	3		7	4	7	
С		7			5	
D		4			2	
E		7	5	2		3
F					3	

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

1. 11 2. 12 3. 13 4. 18

Решение: Сначала немного о графах (небольшая справка, можно пропустить, если знаете).

* * *

Граф — это множество точек (для удобства изображения — на плоскости) и попарно соединяющих их линий (не обязательно прямых). В *графе* важен только факт наличия связи между двумя *вершинами*. От способа изображения этой связи структура *графа* не зависит.

В *графах* не бывает **петель** — рёбер, соединяющих некоторую *вершину* саму с собой. Кроме того, в классическом *графе* не бывает двух различных рёбер, соединяющих одну и ту же пару *вершин*.

Ребро и *вершина* называются **инцидентными** друг другу, если *вершина* является одним из концов *ребра*.

Любому *ребру* инцидентно ровно две *вершины*, а вот *вершине* может быть инцидентно произвольное количество рёбер, это количество и определяет **степень вершины**. *Изолированная вершина* вообще не имеет инцидентных ей рёбер (её *степень* равна 0).

Две *вершины* называются **смежными**, если они являются разными концами одного *ребра* (иными словами, эти *вершины* инцидентны одному *ребру*). Аналогично, два *ребра* называются **смежными**, если они инцидентны одной *вершине*.

Путь в *графе* — это последовательность *вершин* (без повторений), в которой любые две соседние *вершины* смежны. Например, в *графе*, изображённом на рисунке (рис.1), есть два различных *пути* из *вершины* а в *вершину* с: adbc и abc.

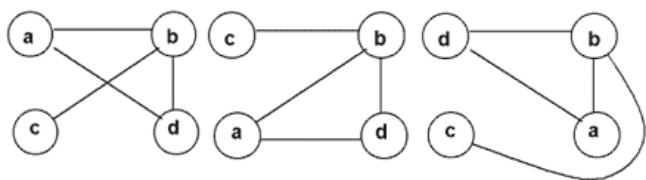


Рис. 1

Вершина **v** **достижима** из *вершины* **u**, если существует *путь*, начинающийся в **u** и заканчивающийся в **v**.

Граф называется **связным**, если все его *вершины* взаимно **достижимы**.

Длина пути — количество рёбер, из которых этот *путь* состоит. Например, *длина* уже упомянутых *путей* adbc и abc (см. рис. 1) — 3 и 2 соответственно.

Расстояние между вершинами u и v — это длина кратчайшего пути от u до v . Из этого определения видно, что расстояние между вершинами a и c в графе на рис. 1 равно 2.

Цикл — это замкнутый путь. Все вершины в цикле, кроме первой и последней, должны быть различны. Например, циклом является путь $abda$ в графе на рис.1.

Ориентированный граф (орграф) — это граф, все рёбра которого имеют направление. Такие направленные рёбра называются **дугами**. На рисунках дуги изображаются стрелочками (см. рис.2).

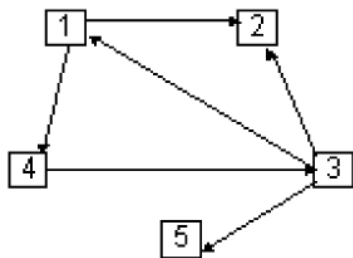


Рис. 2

В отличие от рёбер, дуги соединяют две неравноправные вершины: одна из них называется **началом** дуги (дуга из неё **исходит**), вторая — **концом** дуги (дуга в неё **входит**). Можно сказать, что любое *ребро* — это пара дуг, направленных навстречу друг другу.

Путь в орграфе — это последовательность вершин (без повторений), в которой любые две соседние вершины смежны, причём каждая вершина является одновременно концом одной дуги и началом следующей дуги. «Двигаться» по орграфу можно только в направлениях, заданных стрелками.

Взвешенный (другое название: **размеченный**) граф (или орграф) — это граф (орграф), некоторым элементам которого (вершинам, рёбрам или дугам) сопоставлены числа. Наиболее часто встречаются графы с помеченными рёбрами. Числа-пометки носят различные названия: **вес, длина, стоимость**.

Замечание: Обычный (не взвешенный) граф можно интерпретировать как взвешенный, все рёбра которого имеют одинаковый вес 1.

Длина пути во взвешенном (связном) графе — это сумма длин (весов) тех рёбер, из которых состоит путь. **Расстояние** между вершинами — это, как и прежде, длина кратчайшего пути.

Например, расстояние от вершины a до вершины d во взвешенном графе, изображённом на рис.3, равно 6.

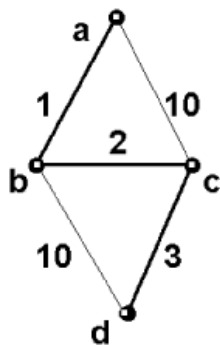


Рис.3

Один из способов представления графов является **матрица смежности**.

Матрица смежности Sm — это квадратная матрица размером $N \times N$ (N — количество вершин в графе), заполненная единицами и нулями по следующему правилу:

Если в графе имеется ребро e , соединяющее вершины u и v , то $Sm[u, v] = 1$, в противном случае $Sm[u, v] = 0$.

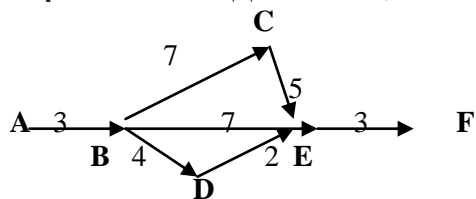
Задать взвешенный граф при помощи матрицы смежности тоже возможно. Необходимо лишь внести небольшое изменение в определение:

Если в графе имеется ребро e , соединяющее вершины u и v , то $Sm[u, v] = \text{ves}(e)$, в противном случае $Sm[u, v] = 0$.

Если у нас имеется матрица смежности, то по ней легко построить граф.

* * *

А теперь о главном. Дана таблица смежности, исходя из нее, можно построить взвешенный граф:



Длина пути во взвешенном (связном) графе — это сумма длин (весов) тех рёбер, из которых состоит путь. У нас три пути: ABEF, ABCEF и ABDEF. Определим каждую длину пути:

ABEF: $3+7+3=13$

ABCEF: $3+7+5+3=18$

ABDEF: $3+4+3+2=12$

Таким образом, кратчайший путь – ABDEF, равный 12.

Ответ: 2

А3. Дан фрагмент таблицы истинности выражения F.

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

Каким из приведённых ниже выражений может быть F?

1. $\neg x_1 \wedge x_2 \wedge \neg x_3 \wedge x_4 \wedge x_5 \wedge \neg x_6 \wedge \neg x_7$

2. $\neg x_1 \vee x_2 \vee \neg x_3 \vee x_4 \vee \neg x_5 \vee \neg x_6 \vee x_7$

3. $x_1 \wedge \neg x_2 \wedge x_3 \wedge \neg x_4 \wedge x_5 \wedge x_6 \wedge \neg x_7$

4. $x_1 \vee \neg x_2 \vee x_3 \vee \neg x_4 \vee \neg x_5 \vee x_6 \vee \neg x_7$

Решение: Для решения задачи, надо помнить следующие правила:

- если в выражении нет скобок, сначала выполняются все операции «НЕ», затем – «И», затем – «ИЛИ», «импликация», и самая последняя – «эквивалентность»;
- таблица истинности выражения определяет его значения при всех возможных комбинациях исходных данных;
- логическая сумма $A \vee B \vee C \vee \dots$ равна 0 (выражение ложно) тогда и только тогда, когда все слагаемые одновременно равны нулю, а в остальных случаях равна 1 (выражение истинно);
- логическое произведение $A \wedge B \wedge C \wedge \dots$ равно 1 (выражение истинно) тогда и только тогда, когда все сомножители одновременно равны единице, а в остальных случаях равно 0 (выражение ложно);
- логическое следование (импликация) $A \rightarrow B$ равна 0 тогда и только тогда, когда из A (посылка) истинна, а B (следствие) ложно;
- эквивалентность $A \equiv B$ равна 1 тогда и только тогда, когда оба значения одновременно равны 0 или одновременно равны 1.

Для каждой строчки нужно подставить заданные значения x_1, x_2, \dots, x_7 во все функции, заданные в ответах, и сравнить результаты с соответствующими значениями F для этих данных, если для какой-нибудь комбинации x_1, x_2, \dots, x_7 результат не совпадает с соответствующим значением F, оставшиеся строчки можно не рассматривать, поскольку для правильного ответа все три результата должны совпасть со значениями функции F. Если подставить значения x_1, x_2, \dots, x_7 в первый вариант ответа, то мы увидим, что все ответы совпадают с функцией F:

$$\neg 1 \wedge 1 \wedge \neg 0 \wedge 1 \wedge 1 \wedge \neg 1 \wedge \neg 1 = 0 \wedge 1 \wedge 1 \wedge 1 \wedge 1 \wedge 0 \wedge 0 = 0$$

$$\neg 1 \wedge 0 \wedge \neg 1 \wedge 0 \wedge 1 \wedge \neg 1 \wedge \neg 0 = 0 \wedge 0 \wedge 0 \wedge 0 \wedge 1 \wedge 0 \wedge 1 = 0$$

$$\neg 0 \wedge 1 \wedge \neg 0 \wedge 1 \wedge 1 \wedge \neg 0 \wedge \neg 0 = 1 \wedge 1 \wedge 1 \wedge 1 \wedge 1 \wedge 1 \wedge 1 = 1$$

Ответ: 1

А4. В каталоге находится 6 файлов:

asc.wma

casting.wmv

last.wma
pasta.wmvx
pasta.wri
vast.wma

Определите, по какой из перечисленных масок из этих 6 файлов будет отображена указанная группа файлов:

casting.wmv
last.wma
pasta.wmvx
vast.wma

1. ?as*.wm? 2. *as?.wm* 3. ?as*.wm* 4. ?as*.w*

Решение: Для решения задачи надо знать следующее:

- данные на дисках хранятся в виде файлов (наборов данных, имеющих имя)
- чтобы было удобнее разбираться со множеством файлов, их объединяют в каталоги (в *Windows* каталоги называются «папками»)
- каталог можно воспринимать как контейнер, в котором размещаются файлы и другие каталоги, которые называются подкаталогами или *вложенными* каталогами (они находятся внутри другого каталога, вложены в него)
- каталоги организованы в многоуровневую (иерархическую) структуру, которая называется «деревом каталогов»
- главный каталог диска (который пользователь видит, «открыв» диск, например, в Проводнике *Windows* или аналогичной программе) называется *корневым* каталогом или «корнем» диска, он обозначается буквой логического диска, за которой следует двоеточие и знак «\» (обратный слэш); например, **A:** – это обозначение корневого каталога диска А
- каждый каталог (кроме корневого) имеет (один единственный!) «родительский» каталог – этот тот каталог, внутри которого находится данный каталог
- полный адрес каталога – это перечисление каталогов, в которые нужно войти, чтобы попасть в этот каталог (начиная с корневого каталога диска); например, **D:\USER\KLASS\PASCAL**
- полный адрес файла состоит из адреса каталога, в котором он находится, символа «\» и имени файла, например, **D:\USER\KLASS\PASCAL\p1.pas**
- маска служит для обозначения (выделения) группы файлов, имена которых имеют общие свойства, например, общее расширение
- в масках, кроме «обычных» символов (допустимых в именах файлов) используются два специальных символа: звездочка «*» и знак вопроса «?»;
- звездочка «*» обозначает любое количество любых символов, в том числе, может обозначать пустую последовательность;
- знак вопроса «?» обозначает ровно один любой символ

Так как во всех выбранных файлах, перед символами «as», находится только один символ, то маска должна начинаться с символа «?». После символов «as» в выбранных файлах стоит разное количество символов. Следовательно, в маске надо поставить четвертым символом – «*». В расширении первые два символа «wm» одинаковые, а после них следует разное количество произвольных символов, следовательно надо поставить в маске последний символ – «*». Итак, подходит для всех выбранных файлов маска - **?as*.wm***.

Ответ: 3

A5. Автомат получает на вход два двузначных шестнадцатеричных числа. В этих числах все цифры не превосходят цифру 6 (если в числе есть цифра больше 6, автомат отказывается работать). По этим числам строится новое шестнадцатеричное число по следующим правилам.

1. Вычисляются два шестнадцатеричных числа – сумма старших разрядов полученных чисел и сумма младших разрядов этих чисел.

2. Полученные два шестнадцатеричных числа записываются друг за другом в порядке возрастания (без разделителей).

Пример. Исходные числа: 66, 43. Поразрядные суммы: А, 9. Результат: 9А.

Определите, какое из предложенных чисел может быть результатом работы автомата.

1. 9F 2. 911 3. 42 4. 7А

Решение: На вход автомату поступают два 16-ных числа, причем каждая цифра в них не превосходит цифру 6. Следовательно, при поразрядном сложении каждая цифра в новом числе не может превосходить 16-ную цифру С (12). Рассмотрим варианты ответов, 9F не подходит, т.к. $F > C$, 911 тоже не подходит, т.к. число трехзначное, 42 – не подходит, т.к. цифры должны быть записаны друг за другом в порядке возрастания. Остается последний вариант – 7А.

Ответ: 4

А6. Ниже представлены две таблицы из базы данных. Каждая строка таблицы 2 содержит информацию о ребёнке и об одном из его родителей. Информация представлена значением поля ID в соответствующей строке таблицы 1. Определите на основании приведённых данных фамилию и инициалы внучки Петровой С.М.

Таблица 1			Таблица 2	
ID	Фамилия И.О.	Пол	ID Родителя	ID Ребёнка
25	Жвания К.Г.	Ж	25	134
49	Черняк А.П.	М	76	49
62	Петрова М.Н.	Ж	76	123
76	Ильченко Т.В.	Ж	82	76
82	Петрова С.М.	Ж	82	96
96	Басис В.В.	Ж	102	76
102	Ильченко В.И.	М	102	96
123	Павлыш Н.П.	Ж	134	49
134	Черняк П.Р.	М	134	123
...

1. Басис В.В.
2. Ильченко Т.В.
3. Павлыш Н.П.
4. Петрова М.Н.

Решение: Петрова С.М. – бабушка, ее код в по таблице 1 равен 82. По таблице 2 находим коды ее детей, т.е. родителей внучки, - это 76 и 96, т.е. или Ильченко Т.В. или Базис В.В.. Теперь для этих родителей по таблице 2 найдем коды их детей. Родителя с кодом 96 в таблице 2 нет, а у Ильченко Т.В. (код 76) – двое детей с кодами 49 и 123 соответственно. Теперь по таблице 1 определяем фамилию внучки. Так как под кодом 49 в таблице 1 находится Черняк А.П. и это мальчик, то следовательно, внучка находится под кодом 123 и это Павлыш Н.П.

Ответ: 3

А7. Дан фрагмент электронной таблицы.

	A	B	C	D
1	1	2	3	
2	5	4	= \$A\$2 + B\$3	
3	6	7	= A3 + B3	

Чему станет равным значение ячейки D1, если в неё скопировать формулу из ячейки C2?

Примечание: знак \$ обозначает абсолютную адресацию.

1. 18
2. 12
3. 14
4. 17

Решение: При выполнении задания надо вспомнить некоторые свойства электронных таблиц:

- адрес ячейки в электронных таблицах состоит из имени столбца и следующего за ним номера строки, например, C15
- формулы в электронных таблицах начинаются знаком = («равно»)
- знаки +, -, *, / и ^ в формулах означают соответственно сложение, вычитание, умножение, деление и возведение в степень
- запись B2:C4 означает диапазон, то есть, все ячейки внутри прямоугольника, ограниченного ячейками B2 и C4:

	A	B	C	D
1				
2				
3				
4				
5				
6				

- например, по формуле =СУММ(B2:C4) вычисляется сумма значений ячеек B2, B3, B4, C2, C3 и C4
- в заданиях ЕГЭ могут использоваться стандартные функции СЧЕТ (количество непустых ячеек), СУММ (сумма), СРЗНАЧ (среднее значение), МИН (минимальное значение), МАКС (максимальное значение)
- функция СРЗНАЧ при вычислении среднего арифметического не учитывает пустые ячейки и ячейки, заполненные текстом; например, после ввода формулы в C2 появится значение 2 (ячейка A2 – пустая):

	A	B	C
1		1	2
2			3
3			=СРЗНАЧ(A1:B2)

функция СЧЕТ(A1:B2) в этом случае выдаст значение 3 (а не 4).

- адреса ячеек (или ссылки на ячейки) бывают относительные, абсолютные и смешанные, вся разница между ними проявляется при копировании формулы в другую ячейку:
 - в *абсолютных* адресах перед именем столбца и перед номером строки ставится знак доллара \$, такие адреса не изменяются при копировании; вот что будет, если формулу =\$B\$2+\$C\$3 скопировать из D5 во все соседние ячейки

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

знак \$ как бы «фиксирует» значение: в абсолютных адресах и имя столбца, и номер строки зафиксированы

- в *относительных* адресах знаков доллара нет, такие адреса при копировании изменяются: номер столбца (строки) изменяется на столько, на сколько отличается номер столбца (строки), где оказалась скопированная формула, от номера столбца (строки) исходной ячейки; вот что будет, если формулу =B2+C3 (в ней оба адреса – относительные) скопировать из D5 во все соседние ячейки:

	C	D	E
4	=A1+B2	=B1+C2	=C1+D2
5	=A2+B3	=B2+C3	=C2+D3
6	=A3+B4	=B3+C4	=C3+D4

- в *смешанных* адресах часть адреса (строка или столбец) – абсолютная, она «зафиксирована» знаком \$, а вторая часть – относительная; относительная часть изменится при копировании так же, как и для относительной ссылки:

	C	D	E
4	=\$B1+B\$3	=\$B1+C\$3	=\$B1+D\$3
5	=\$B2+B\$3	=\$B2+C\$3	=\$B2+D\$3
6	=\$B3+B\$3	=\$B3+C\$3	=\$B3+D\$3
7			

В нашей задаче надо формулу из ячейки D1 скопировать в ячейку C2. В соответствии с вышеизложенным формула =\$A\$2+ B\$3 при копировании преобразуется в формулу =\$A\$2+ C\$3. В ячейке под адресом \$A\$2 находится число 5, а значение ячейки под адресом C\$3 вычисляется по формуле: =A3+B3=6+7=13. Отсюда значение ячейки D1 будет равно: \$A\$2+ C\$3=5+13=18.

Ответ: 1

А8. Производится одноканальная (моно) цифровая звукозапись. Значение сигнала фиксируется 48 000 раз в секунду, для записи каждого значения используется 32 бит. Запись длится 4 минуты, её результаты записываются в файл, сжатия данных не производится. Какая из приведённых ниже величин наиболее близка к размеру полученного файла?

1. 44 Мбайт 2. 87 Мбайт 3. 125 Мбайт 4. 175 Мбайт

Решение: При решении задач на кодирование звук надо знать следующее:

Звук-это колебания воздуха. Аналоговый сигнал - это звук, преобразованный в колебания электрического тока. При кодировании звука аналоговый сигнал преобразуется в оцифрованный сигнал с помощью дискретизации. Непрерывный сигнал преобразуется в набор дискретных значений, каждое из которых представляется двоичным числом.

Частота дискретизации (f) - количество измерений амплитуды аналогового сигнала в единицу времени.

Глубина кодирования (r) - сколько бит выделяется на кодирование значений амплитуды.

Чем больше частота дискретизации (меньше шаг дискретизации) и больше разрешение, тем качественнее будет аудиозапись.

Объем звукозаписи V (размер аудиофайла) определяют по формуле: $V = f \times r \times t \times a$

где f - частота дискретизации, r -разрешение, t -время, a -коэффициент (для монозвука $a=1$, для стерео $a=2$).

Дано: $f = 48000$ раз в сек (Гц)

$r = 32$ бита

$t = 4$ мин $= 4 \times 60 = 240$ сек

$a = 1$

$V = f \times r \times t \times a = 48000 \times 32 \times 240 \times 1$

Так как ответ в мегабайтах, то надо это выражение поделить на 8×2^{20} , получим:

$(48000 \times 32 \times 240 \times 1) / (8 \times 2^{20}) = (2^{10} \times 3000 \times 15) / 2^{20} = 43,9$ Мбайт

Ответ: 1

А9. Для кодирования некоторой последовательности, состоящей из букв А, Б, В, Г и Д, используется неравномерный двоичный код, позволяющий однозначно декодировать полученную двоичную последовательность. Вот этот код: А – 00, Б – 01, В – 100, Г – 101, Д – 110. Можно ли сократить для одной из букв длину кодового слова так, чтобы код по-прежнему можно было декодировать однозначно? Коды остальных букв меняться не должны. Выберите правильный вариант ответа.

1. для буквы Д – 11
2. это невозможно
3. для буквы Г – 10
4. для буквы Д – 10

Решение: **Кодирование** – это перевод информации с одного языка на другой (запись в другой системе символов, в другом алфавите), обычно кодированием называют перевод информации с «человеческого» языка на формальный, например, в двоичный код, а декодированием – обратный переход.

Один символ исходного сообщения может заменяться одним символом нового кода или несколькими символами, а может быть и наоборот – несколько символов исходного сообщения заменяются одним символом в новом коде (китайские иероглифы обозначают целые слова и понятия).

Кодирование может быть *равномерное* и *неравномерное*. При равномерном кодировании все символы кодируются кодами равной длины, а при неравномерном кодировании разные символы могут кодироваться кодами разной длины, это затрудняет декодирование.

Закодированное сообщение можно однозначно декодировать с начала, если выполняется **условие Фано**: *никакое кодовое слово не является началом другого кодового слова.*

Закодированное сообщение можно однозначно декодировать с конца, если выполняется **обратное условие Фано**: *никакое кодовое слово не является окончанием другого кодового слова.*

Условие Фано – это достаточное, но не необходимое условие однозначного декодирования.

Для однозначного декодирования достаточно, чтобы выполнялось условие Фано или обратное условие Фано.

Проверяем последовательно варианты 1, 3, 4. Если ни один вариант не подойдет, то можно выбрать вариант 2 - это невозможно.

Вариант 1. Для буквы Д – 11. Этот вариант удовлетворяет обоим условиям Фано, т.е. 11 не является началом и не является окончанием другого кодового слова.

Решение варианта демо версии ЕГЭ 2013 года (части А и В)

Учитель информатики Батракова Л.В

Вариант 3. Для буквы Г – 10. Этот вариант не подходит, т.к. 10 является началом кодовых слов В – 100, Г – 101 и окончанием кодового слова Д – 110.

Вариант 4. Для буквы Д – 10. Эта ситуация аналогична варианту 3, следовательно не подходит.

Ответ: 1

A10. Для какого из приведённых чисел X истинно логическое условие:

$\neg ((X \text{ кратно } 2) \rightarrow (X \text{ кратно } 4))$?

1. 7 2. 8 3. 10 4. 12

Решение: Если все выражение $\neg ((X \text{ кратно } 2) \rightarrow (X \text{ кратно } 4))$ - истинно, значит выражение $(X \text{ кратно } 2) \rightarrow (X \text{ кратно } 4)$ – ложно. Из свойства импликации известно, что импликация ложна, если из истины следует ложь. Следовательно, выражение $(X \text{ кратно } 2)$ – истинно, а выражение $(X \text{ кратно } 4)$ – ложно. Этому соответствует в ответе число 10.

Ответ: 3

A11. В некоторой стране автомобильный номер длиной 5 символов составляют из заглавных букв (задействовано 30 различных букв) и любых десятичных цифр в любом порядке.

Каждый такой номер в компьютерной программе записывается минимально возможным и одинаковым целым количеством байт (при этом используют посимвольное кодирование и все символы кодируются одинаковым и минимально возможным количеством бит).

Определите объём памяти, отводимый этой программой для записи 50 номеров.

1. 100 байт 2. 150 байт 3. 200 байт 4. 250 байт

Решение:

Дано: мощность алфавита N - $30 + 10 = 40$ символов. По формуле Хартли $N \leq 2^i$ определяем количество бит i на символ - $40 \leq 64 = 2^6$, т.е. i = 6.

Длина номера 5 символов, следовательно на весь номер отводится $6 \times 5 = 30$ бит или 4 байта.

Объём памяти, отводимый для записи 50 номеров равен: $4 \times 50 = 200$ байт.

Ответ: 3

A12. 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
for (i=0; i<=10; i++) A[i]=i-1; for (i=1; i<=10; i++) A[i-1]=A[i]; A[10]=10;	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

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: В программе два цикла. После выполнения этих циклов элементы массива будут заполнены следующим образом:

Индексы массива А	0	1	2	3	4	5	6	7	8	9	10
Значения после первого цикла	-1	0	1	2	3	4	5	6	7	8	9
Значения после второго цикла	0	1	2	3	4	5	6	7	8	9	10

Из таблицы видно, все элементы окажутся равны своим индексам.

Ответ: 2

A13. Сколько клеток лабиринта соответствуют требованию, что, начав движение в ней и выполнив предложенную программу, РОБОТ уцелеет и остановится в закрашенной клетке (клетка A1)?

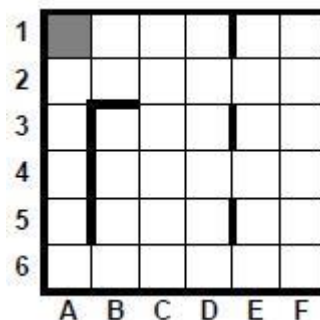
НАЧАЛО ПОКА < слева свободно ИЛИ сверху свободно >
свободно >

ЕСЛИ < слева свободно > ТО влево

ИНАЧЕ вверх

КОНЕЦ ЕСЛИ

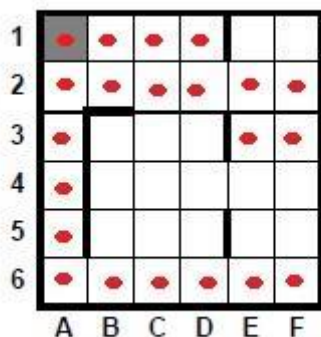
КОНЕЦ ПОКА КОНЕЦ



1. 8 2. 12 3. 17 4. 21

Решение: Программа состоит из цикла со сложным условием ПОКА < слева свободно ИЛИ сверху свободно > и условия ЕСЛИ < слева свободно > ТО влево ИНАЧЕ вверх. Рассматривая все клетки лабиринта, надо выбрать те из них, которые удовлетворяют этим условиям.

Робот уцелеет, если начнет движение из клеток, показанных на рисунке. Всего таких клеток 21.



Ответ: 4

ЧАСТЬ В

В1. У исполнителя Арифметик две команды, которым присвоены номера:

1. прибавь 2,

2. умножь на 3.

Первая из них увеличивает число на экране на 2, вторая утраивает его.

Например, **21211** – это программа

умножь на 3

прибавь 2

умножь на 3

прибавь 2

прибавь 2,

которая преобразует число 1 в число 19.

Запишите порядок команд в программе преобразования **числа 3 в число 69**, содержащей не более 5 команд, указывая лишь номера команд. Если таких программ более одной, то запишите любую из них.

Решение: Нужно увеличить число (с 3 до 69), для этого в большинстве случаев умножение эффективнее сложения, поэтому нужно постараться максимально использовать умножение, а сложение – только в крайних случаях. Попробуем решить задачу «**обратным ходом**», начав с числа 69, если число делится на 3, то используется команда 2, если нет, то 1:

1) очевидно, что последней командой может быть умножение на 3 (69 на 3 делится), поэтому последняя команда – **2. умножь на 3**, над стрелкой записан номер команды:

$$\dots 23 \xrightarrow{2} 69$$

2) число 23 не делится на 3, а 21 – делится на 3, поэтому предыдущей командой должна быть команда - **1. прибавь 2:**

$$\dots 21 \xrightarrow{1} 23 \xrightarrow{2} 69$$

3) число 21 делится на 3, поэтому следующая команда - 2:

$$\dots 7 \xrightarrow{2} 21 \xrightarrow{1} 23 \xrightarrow{2} 69$$

4) число 7 не делится на 3, но применив дважды команду 1, получим исходное значение – 3:

$$3 \xrightarrow{1} 5 \xrightarrow{1} 7 \xrightarrow{2} 21 \xrightarrow{1} 23 \xrightarrow{2} 69$$

Таким образом, правильный ответ – 11212, эта программа состоит из 5 команд.

Ответ: **11212**

В2. Определите значение переменной **с** после выполнения следующего фрагмента программы (*записанного ниже на разных языках программирования*). Ответ запишите в виде целого числа.

Бейсик	Паскаль
<pre> a = 30 b = 14 a = a - 2 * b IF a > b THEN c = b + 2 * a ELSE c = b - 2 * a ENDIF </pre>	<pre> a := 30; b := 14; a := a - 2 * b; if a > b then c := b + 2 * a else c := b - 2 * a; </pre>
Си	Алгоритмический
<pre> a = 30; b = 14; a = a - 2 * b; if (a > b) c = b + 2 * a; else c = b - 2 * a; </pre>	<pre> a := 30 b := 14 a := a - 2 * b если a > b то c := b + 2 * a иначе c := b - 2 * a все </pre>

Решение: Здесь все просто. Пошагово выполняйте алгоритм. Начальные значения: $a=30$, $b=14$. После выполнения 3-его оператора значение переменной a изменится: $a=30-2*14=2$. Так как a стало меньше b , то выполняется оператор после `else`, т.е. $c:=b-2*a=14-2*2=10$.

Ответ: 10

В3. Дан фрагмент электронной таблицы.

	A	B	C
1	2	4	
2	$=(B1 - A1)/2$	$= 2 - A1/2$	$=(C1 - A1)*2 - 4$

Какое целое число должно быть записано в ячейке C1, чтобы построенная после выполнения вычислений диаграмма по значениям диапазона ячеек A2 : C2 соответствовала рисунку?

Известно, что все значения диапазона, по которым построена диаграмма, имеют один и тот же знак.



Решение: Вычисляем значения в ячейках A2 и B2:

$$A2=(4-2)/2=1 \text{ и } B2=2-2/2=1$$

На диаграмме $A2=B2=1$, тогда $C2=2$.

По формуле, указанной в ячейке C2 составляем уравнение и решаем его.

$$C2=(C1-2)*2 - 4 = 2 \rightarrow (C1-2)*2=6 \rightarrow C1-2=3 \rightarrow C1=5$$

Ответ: 5

В4. 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: Код слова имеет 4 или 5 символов. Используя формулу Хартли, получим $2^4=16$ - количество 4-битных символов и $2^5=32$ - количество 5-битных символов. Всего символов: $16+32=48$

Ответ: 48

В5. Определите, что будет напечатано в результате выполнения программы (записанной ниже на разных языках программирования).

Бейсик	Паскаль
<pre> DIM N, S AS INTEGER N = 0 S = 0 WHILE S <= 35 N = N + 1 S = S + 4 WEND PRINT N </pre>	<pre> var n, s: integer; begin n := 0; s := 0; while s <= 35 do begin n := n + 1; s := s + 4 end; write(n) end. </pre>
C++	Алгоритмический
<pre> #include<stdio.h> void main() { int n, s; n = 0; s = 0; while (s <= 35) { n = n + 1; s = s + 4; } printf("%d", n); } </pre>	<pre> алг нач цел n, s n := 0 s := 0 нц пока s <= 35 n := n + 1 s := s + 4 кц вывод n кон </pre>

Решение варианта демо версии ЕГЭ 2013 года (части А и В)

Учитель информатики Батракова Л.В

Решение: Для решения можно использовать «ручную прокрутку» программы, то есть, выполнить вручную все действия. Наиболее удобно и наглядно это получается при использовании таблицы, где в столбцах показаны изменения переменных при выполнении операторов в цикле пока не выполнится условие выхода из цикла:

n	s
0	0
1	4
2	8
3	12
4	16
5	20
6	24
7	28
8	32
9	36

Будет выведено значение n=9.

Ответ: 9

В6. Алгоритм вычисления значения функции $F(n)$, где n – натуральное число, задан следующими соотношениями:

$$F(1) = 1$$

$$F(n) = F(n-1) * n, \text{ при } n > 1$$

Чему равно значение функции $F(5)$?

В ответе запишите только натуральное число.

Решение: Это алгоритм рекурсивной функции вычисления $N!$ (Надеюсь, что вы это увидели ;)). А $5! = 120$.

Ответ: 120

В7. Запись десятичного числа в системах счисления с основаниями 3 и 5 в обоих случаях имеет последней цифрой 0. Какое минимальное натуральное десятичное число удовлетворяет этому требованию?

Решение: Для решения задачи нужно рассмотреть числа в троичной и пятеричной системах, оканчивающиеся на ноль. Надо помнить, что алфавит троичной системы состоит из цифр $\{0, 1, 2\}$, а в пятеричной – $\{0, 1, 2, 3, 4\}$.

$$(10)_3 = 3; (20)_3 = 6; (100)_3 = 9; (110)_3 = 12; (120)_3 = 15;$$

$$(10)_5 = 5; (20)_5 = 10; (30)_5 = 15$$

Минимальное натуральное десятичное число удовлетворяет этому требованию – 15.

Ответ: 15

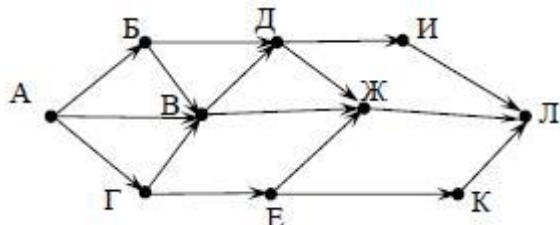
В8. Ниже на четырёх языках записан алгоритм. Получив на вход число x , этот алгоритм печатает два числа: a и b . Укажите наименьшее из таких чисел x , при вводе которых алгоритм печатает сначала 2, а потом 21.

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>
Си	Алгоритмический
<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 нц пока x>0 a:=a+1 b:=b*mod(x,10) x:=div(x,10) кц вывод a, nc, b кон </pre>

Решение: Переменная **a** определяет, сколько раз выполняется цикл. Так как по условию $a=2$, то число **x** должно быть двузначным. В переменной **b** вычисляется произведение цифр числа **x**. Оно по условию должно быть равно 21. Следовательно, число **x** должно состоять из цифр 3 и 7, т.е. это числа 37 или 73. В ответе надо указать наименьшее число, т.е. 37.

Ответ: 37

В9. На рисунке – схема дорог, связывающих города А, Б, В, Г, Д, Е, Ж, И, К, Л. По каждой дороге можно двигаться только в одном направлении, указанном стрелкой. Сколько существует различных путей из города А в город Л?



Решение: Эту задачу можно решать несколькими способами (см. сайт К. Полякова). Самый простой способ – подстановки:

- 1) начнем считать количество путей с конца маршрута – с города Л
- 2) будем обозначать через N_X количество различных путей из города А в город X
- 3) общее число путей обозначим через N
- 4) по схеме видно, что $N_B = N_G = 1$
- 5) очевидно, что если в город X можно приехать только из Y, Z, то $N_X = N_Y + N_Z$, то есть нужно сложить число путей, ведущих из А во все города, откуда можно приехать в город X
- 6) поскольку в Л можно приехать из И, Ж или К, поэтому

$$N = N_L = N_I + N_J + N_K$$
- 7) в город И можно приехать только из Д, поэтому $N_I = N_D$
- 8) в город Ж можно приехать только из Е, Д и В, поэтому

$$N_J = N_E + N_D + N_B$$
- 9) в город К можно приехать только из Е, поэтому $N_K = N_E$
- 10) подставляем результаты пп. 7, 8 и 9 в формулу п. 6:

$$N_L = N_B + 2N_E + 2N_D$$
- 11) в город Д можно приехать только из Б и В, поэтому

$$N_D = N_B + N_V$$
- 12) в город Е можно приехать только из Г, поэтому $N_E = N_G$
- 13) в город В можно приехать только из Б, А и Г, поэтому

$$N_B = N_B + N_A + N_G$$
- 14) подставляем результаты пп. 11, 12 и 13 в формулу п. 10:

$$N_L = 3N_B + 2N_B + 3N_G + N_A$$
- 15) по схеме видно, что $N_A = N_B = N_G = 1$, кроме того, $N_B = N_A + N_B + N_G = 3$
- 16) окончательно $N = 3N_B + 2N_B + 3N_G + N_A = 3 \cdot 1 + 2 \cdot 3 + 3 \cdot 1 + 1 = 13$

Ответ: 13

В10. Документ объёмом 20 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^{20} бит в секунду;
- объём сжатого архиватором документа равен 20% исходного;
- время, требуемое на сжатие документа, – 5 секунд, на распаковку – 1 секунда?

В ответе напишите букву А, если быстрее способ А, или Б, если быстрее способ Б. Сразу после буквы напишите число, обозначающее, на сколько секунд один способ быстрее другого.

Так, например, если способ Б быстрее способа А на 23 секунды, в ответе нужно написать Б23.

Единиц измерения «секунд», «сек.», «с.» к ответу добавлять не нужно.

Решение: Дано: $I = 20 \text{ Мбайт} = 20 \times 2^{23} \text{ бит}$ – объём документа

$V = 2^{20}$ бит/с – скорость передачи данных

$I_{сж} = 20 \text{ Мбайт} \times 0.2 = 4 \text{ Мбайт} = 4 \times 2^{23} \text{ бит}$ – объём сжатого документа

$t = 5 + 1 = 6$ (с) – время, требуемое для сжатия и распаковки документа.

1. Определяем время, необходимое для передачи по каналу связи без использования архиватора:

$$t_B = I/V = 20 \times 2^{23} / 2^{20} = 20 \times 2^3 = 20 \times 8 = 160 \text{ (с)}$$

2. Определяем время, необходимое для сжатия, передачи и распаковки документа:

$$t_A = t + 4 \times 2^{23} / 2^{20} = 6 + 4 \times 8 = 6 + 32 = 38 \text{ (с)}$$

3. Определяем на сколько способ А быстрее способа Б: $t_B - t_A = 160 - 38 = 122 \text{ с}$

Ответ: А122

В11. 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 заданному IP-адресу узла и маске.

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

IP-адрес узла: 217.19.128.131

Маска: 255.255.192.0

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

А	В	С	Д	Е	Ф	Г	Н
0	16	19	64	128	131	192	217

Пример.

Пусть искомый IP-адрес: 192.168.128.0, и дана таблица

А	В	С	Д	Е	Ф	Г	Н
128	168	255	8	127	0	17	192

В этом случае правильный ответ будет записан в виде: **НВАФ**

Решение: Каждый компьютер, подключенный к сети Интернет, должен иметь собственный адрес, который называют IP-адресом (IP = *Internet Protocol*). IP-адрес состоит из четырех чисел, разделенных точками, каждое из этих чисел находится в интервале 0...255, например: **128.168.255.210**.

Поскольку $255 = 11111111_2$, все части IP-адреса узла, для которых маска равна 255, входят в IP-адрес сети без изменений (они полностью относятся к номеру сети), поскольку $0 = 00000000_2$, все части IP-адреса узла, для которых маска равна 0, в IP-адресе сети заменяются нулями (они полностью относятся к номеру узла в сети). Таким образом, мы почти определили адрес сети, он равен 217.19.X.0, где X придется определять дополнительно.

Переведем в двоичную систему третью часть IP-адреса и маски

Маска $192_{10} = 1100\ 0000_2$

IP-адрес $128_{10} = 1000\ 0000_2$

Применим поразрядную конъюнкцию, получим $1000\ 0000_2 = 128_{10}$

Мы определили полный адрес сети, он равен 217.19.128.0

По таблице находим ответ: **НСЕА**

Ответ: НСЕА

В12. 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>Фрегат</i> <i>Эсминец</i>	3400
<i>Фрегат</i> & <i>Эсминец</i>	900
<i>Фрегат</i>	2100

Какое количество страниц (в тысячах) будет найдено по запросу *Эсминец*?

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

Решение: Для решения задачи построим диаграмму Эйлера-Венна для двух множеств – фрегаты и эсминцы.



Число сайтов, удовлетворяющих запросу Фрегат|Эсминец - это область $1+2+3=3400$.

Число сайтов, удовлетворяющих запросу Фрегат&Эсминец - это область область $3=900$.

Число сайтов, удовлетворяющих запросу Фрегат - область $1+3= 2100$.

Надо подсчитать количество запросов для области "Эсминец", т.е. $2+3$:

$$2+3 = (1+2+3) - (1+3) + 3 = 3400 - 2100 + 900 = \mathbf{2200}$$

Ответ: **2200**

В13. У исполнителя Удвоитель две команды, которым присвоены номера:

1. прибавь 1,

2. умножь на 2.

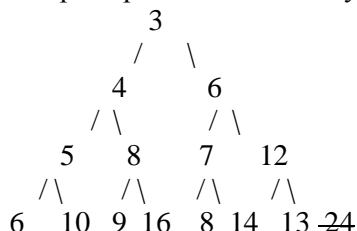
Первая из них увеличивает на 1 число на экране, вторая удваивает его.

Программа для Удвоителя – это последовательность команд.

Сколько есть программ, которые число 3 преобразуют в число 23?

Решение:

Построим дерево решения. Влево будем увеличивать число на 1, а вправо - удваивать число.



Дальше продолжите сами. Главное - идея. Не забывайте, что вам надо получить число **23**. Чтобы не строить громоздкое дерево можно записывать в строчку на каждом шаге новые значения ≤ 23 . (Вариант тоже длинный, но точно приведет к верному результату.)

Например:

- 1) 3
- 2) 4 6
- 3) 5 8 7 12
- 4) 6 10 9 16 8 14 13
- 5) 7 12 11 20 10 18 17 9 16 15 14
- 6) 8 14 13 12 22 21 11 20 19 18 10 18 17 16 15
- 7) 9 16 15 14 13 **23** 22 12 22 21 20 19 11 20 19 18 17 16

После этого шага, когда мы получили первое число равное 23, можно сказать, что все числа, большие 11 дадут нам одно решение за счет выполнения команды 1 (прибавить 1) их 15. Получили 16 решений. Остается определить, сколько решений можно получить из чисел 9 и 11.

Число 11 даст 2 решения:

- 1) 11
- 2) 12 22
- 3) 13 **23**

Из 13 командой 1 получим 23.

Число 9 даст 4 решения:

- 1) 9
- 2) 10 18
- 3) 11 20 19
- 4) 12 22 21 20 из всех этих чисел можно получить одно решение командой 1.

Итого получается 22 решения.

Ответ: 22

В14. Определите, какое число будет напечатано в результате выполнения следующего алгоритма (для Вашего удобства алгоритм представлен на четырёх 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) END FUNCTION </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>
Сп	Алгоритмический
<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: Программа ищет значение **t**, при котором функция **F(t)** принимает минимальное значение на интервале от **a** до **b**.

Заданная функция $F(x) = 3(x-8)(x-8)$ имеет корни в точках $x_1 = 8$, $x_2 = 8$.

График этой функции – парабола, оси которой направлены вверх, поэтому функция имеет минимум.

Парабола симметрична относительно вертикальной прямой, проходящей через вершину, поэтому абсцисса вершины – это среднее арифметическое корней:

$$x_{\min} = \frac{8+8}{2} = 8$$

Ответ: 8

В15. Сколько существует различных наборов значений логических переменных $x_1, x_2, x_3, x_4, y_1, y_2, y_3, y_4$, которые удовлетворяют всем перечисленным ниже условиям?

$$(x1 \rightarrow x2) \wedge (x2 \rightarrow x3) \wedge (x3 \rightarrow x4) = 1$$

$$(\neg y1 \vee y2) \wedge (\neg y2 \vee y3) \wedge (\neg y3 \vee y4) = 1$$

$$(y1 \rightarrow x1) \wedge (y2 \rightarrow x2) \wedge (y3 \rightarrow x3) \wedge (y4 \rightarrow x4) = 1$$

В ответе **не нужно** перечислять все различные наборы значений переменных $x_1, x_2, x_3, x_4, y_1, y_2, y_3, y_4$, при которых выполнена данная система равенств. В качестве ответа Вам нужно указать количество таких наборов.

Решение: Для решения задач такого типа, надо знать операции, выполняемые над логическими переменными.

- условные обозначения логических операций

$\neg A, \bar{A}$ не A (отрицание, инверсия)

$A \wedge B, A \cdot B$ A и B (логическое умножение, конъюнкция)

$A \vee B, A + B$ A или B (логическое сложение, дизъюнкция)

$A \rightarrow B$ импликация (следование)

$A \leftrightarrow B, A \equiv B$ эквиваленция (эквивалентность, равносильность)

- операцию «импликация» можно выразить через «ИЛИ» и «НЕ»:

$$A \rightarrow B = \neg A \vee B \text{ или в других обозначениях } A \rightarrow B = \bar{A} + B$$

- операцию «эквиваленция» также можно выразить через «ИЛИ» и «НЕ»:

$$A \leftrightarrow B = \neg A \wedge \neg B \vee A \wedge B \text{ или в других обозначениях } A \leftrightarrow B = \bar{A} \cdot \bar{B} + A \cdot B$$

- если в выражении нет скобок, сначала выполняются все операции «НЕ», затем – «И», затем – «ИЛИ», потом – «импликация», и самая последняя – «эквиваленция»
- логическое произведение $A \cdot B \cdot C \dots$ равно 1 (выражение истинно) только тогда, когда все сомножители равны 1 (а в остальных случаях равно 0)
- логическая сумма $A + B + C \dots$ равна 0 (выражение ложно) только тогда, когда все слагаемые равны 0 (а в остальных случаях равна 1)
- правила преобразования логических выражений (законы алгебры логики):

Закон	Для И	Для ИЛИ
двойного отрицания	$\overline{\bar{A}} = A$	
исключения третьего	$A \cdot \bar{A} = 0$	$A + \bar{A} = 1$
исключения констант	$A \cdot 1 = A; A \cdot 0 = 0$	$A + 0 = A; A + 1 = 1$
повторения	$A \cdot A = A$	$A + A = A$
поглощения	$A \cdot (A + B) = A$	$A + A \cdot B = A$
переместительный	$A \cdot B = B \cdot A$	$A + B = B + A$
сочетательный	$A \cdot (B \cdot C) = (A \cdot B) \cdot C$	$A + (B + C) = (A + B) + C$
распределительный	$A + B \cdot C = (A + B) \cdot (A + C)$	$A \cdot (B + C) = A \cdot B + A \cdot C$
де Моргана	$\overline{A \cdot B} = \bar{A} + \bar{B}$	$\overline{A + B} = \bar{A} \cdot \bar{B}$

Первые два уравнения независимы друг от друга (в первое входят только x_1, x_2, \dots, x_4 , а во второе – только y_1, y_2, \dots, y_4). Третье уравнение связывает первые два, поэтому можно поступить так:

- найти решения первого уравнения
- найти решения второго уравнения
- найти множество решений первых двух уравнений
- из множества решений первых двух уравнений выкинуть те, которые не удовлетворяют последнему уравнению

Найдем решения первого уравнения. Каждая из логических переменных x_1, x_2, \dots, x_4 может принимать только два значения: «ложь» (0) и «истина» (1), поэтому решение первого уравнения можно записать как битовую цепочку длиной 4 бита: например, 0011 означает, что

$$x_1 = x_2 = 0 \text{ и } x_3 = x_4 = 1$$

Так как импликация $x_1 \rightarrow x_2$ ложна только для $x_1 = 1$ и $x_2 = 0$, поэтому битовая цепочка, представляющая собой решение первого уравнения, не должна содержать сочетания «10»; это дает такие решения (других нет!):

$$(x_1, x_2, x_3, x_4) = 0000 \quad 0001 \quad 0011 \quad 0111 \quad 1111$$

Так как $A \rightarrow B = \neg A \vee B$, то второе уравнение можно представить в виде:

$$(y1 \rightarrow y2) \wedge (y2 \rightarrow y3) \wedge (y3 \rightarrow y4) = 1$$

Оно полностью совпадает по форме с первым, поэтому все его решения:

$$(y_1, y_2, y_3, y_4) = 0000 \quad 0001 \quad 0011 \quad 0111 \quad 1111$$

поскольку первые два уравнения независимы друг от друга, система из первых двух уравнений имеет $5 \cdot 5 = 25$ решений: каждому решению первого соответствует 5 разных комбинаций переменных y_1, y_2, \dots, y_4 , которые решают второе, и наоборот, каждому решению второго соответствует 5 разных комбинаций переменных x_1, x_2, \dots, x_4 , которые решают первое:

$$\begin{array}{l} (y_1, y_2, y_3, y_4) = \\ (x_1, x_2, x_3, x_4) = \end{array} \begin{array}{ccccc} 0000 & 0001 & 0011 & 0111 & 1111 \\ 0000 & 0000 & 0000 & 0000 & 0000 \\ 0001 & 0001 & 0001 & 0001 & 0001 \\ 0011 & 0011 & 0011 & 0011 & 0011 \\ 0111 & 0111 & 0111 & 0111 & 0111 \\ 1111 & 1111 & 1111 & 1111 & 1111 \end{array}$$

Проверим, какие ограничения накладывает третье уравнение.

Так как импликация $y_i \rightarrow x_i$ ложна только для $y_i = 1$ и $x_i = 0$, следовательно, такая комбинация запрещена, потому что нарушает третье уравнение. Из множества решений удаляем все такие решения и получим:

$$\begin{array}{l} (y_1, y_2, y_3, y_4) = \\ (x_1, x_2, x_3, x_4) = \end{array} \begin{array}{ccccc} 0000 & 0001 & 0011 & 0111 & 1111 \\ 0000 & & & & \\ 0001 & 0001 & & & \\ 0011 & 0011 & 0011 & & \\ 0111 & 0111 & 0111 & 0111 & \\ 1111 & 1111 & 1111 & 1111 & 1111 \end{array}$$

Итак, остается одно решение при $(y_1, y_2, y_3, y_4) = 1111$, два решения при $(y_1, y_2, y_3, y_4) = 0111$, три решения при $(y_1, y_2, y_3, y_4) = 0011$, четыре решения при $(y_1, y_2, y_3, y_4) = 0001$ и 5 решений при $(y_1, y_2, y_3, y_4) = 0000$

Всего решений $1+2+3+4+5=15$.

Ответ: **15**