



## Новые “вилы” от Министерства образования, или Как решать принципиально новые задачи

**О.Б. Богомолова,**  
д. п. н., учитель информатики  
и математики ГБОУ СОШ № 1360,  
Восточный округ г. Москвы

**Д.Ю. Усенков,**  
ст. н. с. Института  
информатизации  
образования  
Российской академии  
образования, Москва

**А.Ремизова,**  
ученица 11-го “А” класса  
школы № 1360  
г. Москвы

► Наша школа недавно приняла на себя роль “экспериментальной площадки” для апробации новых заданий ЕГЭ по математике и информатике. И в конце октября 2014 года нашим школьникам было предложено написать пробную диагностическую работу в формате ЕГЭ по информатике, в которой многие задания были не просто усложненными, но и принципиально новыми по формулировкам.

Насколько оправданно еще более усложнять задания ЕГЭ при том, что результативность сдачи экзамена и на прежних заданиях в масштабе страны оказалась, мягко говоря, невысокой, — вопрос спорный. Может быть, для того, чтобы

закрыть дорогу в профильные вузы еще большему количеству одиннадцатиклассников? Однако наша задача теперь — предполагая, что такие задания могут встретиться на реальном ЕГЭ, научить ребят решать и их.

Ниже приводятся решения задач одного из вариантов пробной диагностической работы, благо на листах с этими заданиями нигде не указан запрет на их публикацию.

Итак...

1. Укажите наибольшее число, двоичная запись которого содержит ровно три значащих нуля и две единицы. Ответ запишите в десятичной системе счисления.

*Решение*

1) Значащие нули не могут располагаться в самом начале числа, — значит, число должно начинаться хотя бы с одной единицы.

2) Вспомним формулу перевода двоичного числа в десятичное: чем левее в двоичном числе единица, тем бóльшая степень двойки входит в качестве слагаемого в эту запись. Следовательно, максимальное значение двоичного числа получится, если все единицы будут “собраны” в старших разрядах.

3) Требуется, чтобы в записи числа было ровно три значащих нуля и две единицы. Следовательно, число — 5-разрядное. Если все единицы размещаются в старших разрядах, то это число  $11000_2$ .

4) Ответ требуется записать в десятичной форме. Переведем полученное двоичное число в десятичное:

$$11000_2 = 2^4 + 2^3 = 16 + 8 = 24.$$

*Ответ:* 24.

2. Между населенными пунктами А, В, С, D, E, F, Z построены дороги с односторонним движением. В таблице указана протяженность каждой дороги. Отсутствие числа в таблице означает, что прямой дороги между пунктами нет. Например, из А в В есть дорога длиной 4 км, а из В в А дороги нет.

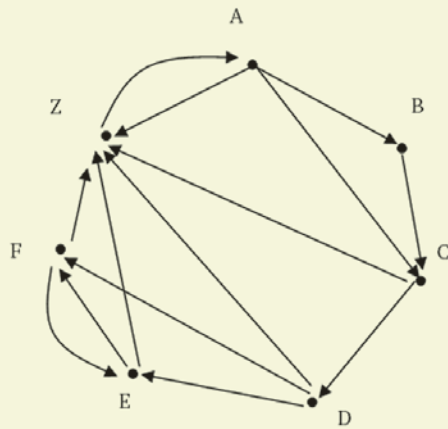
	А	В	С	D	E	F	Z
А		4	6				30
В			3				
С				11			27
D					4	7	10
E						4	8
F					5		2
Z	29						

Сколько существует таких маршрутов из А в Z, которые проходят через 6 и более населенных пунктов? Пункты А и Z при подсчете учитывать. Два раза проходить через один пункт нельзя.

*Решение*

Речь идет о построении графа по заданной табличной форме (как ранее в аналогичных задачах А2), но графа *ориентированного*, в котором ребра обладают направлениями. Их нужно обозначать не отрезками, а стрелками. В случае же, если в таблице заданы и прямой, и обратный пути (как между пунктами А и Z), надо их изобразить двумя отдельными соответствующими стрелками. (Если бы в задаче требовалось определить длину пути, каждая стрелка должна была бы изображаться со своим *весом* — обозначением длины пути.) Точки (вершины графа) располагаем произвольно, не учитывая реальное их расположение, т.е. граф отражает лишь условную “карту дорог”.

1) Строим требуемый граф:



2) Начинаем подсчитывать пути. При этом помним, что через один и тот же пункт в пределах одного какого-то пути нельзя проходить дважды. (Можно при решении задачи на бумаге прорисовывать каждый путь своим цветом при помощи фломастера.) Например:

— AZ: путь напрямую из пункта А в пункт Z существует, но проходит только через два пункта, а не через шесть и более, — значит, он нам не подходит. (Обратный путь из Z в А вообще не учитываем, так как по условию задачи ищутся только пути из А в Z.)

— ABCDEFZ: путь через все вершины графа — он подходит, так как проходит через семь вершин и ни одна из них не посещается дважды.

Аналогичным способом просматриваем остальные пути и выбираем среди них нужные.

Сложно? Есть риск что-то пропустить? Вспомним тогда прежнюю задачу типа В9, где по ориентированному графу требовалось определить количество всех путей из одного города в другой. Ведь, по сути, это та же самая задача!

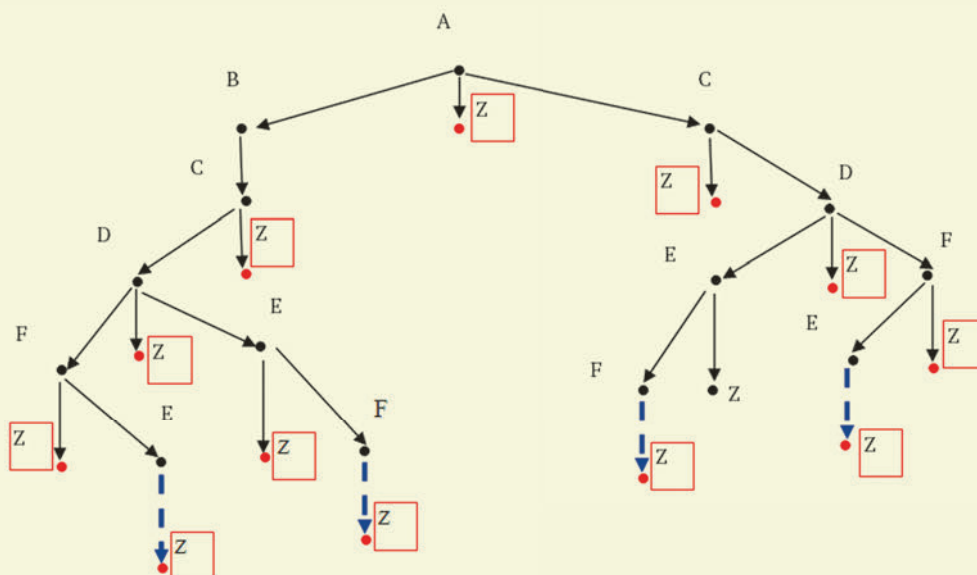
3) Рисуем дерево всех возможных путей из А в Z, при этом конечные точки маршрута (Z) выделяем цветом и рамкой. При этом важно не забыть о запрете на повторное посещение вершин. Это означает, что обратный путь FE нельзя рисовать в тех ветвях дерева, где уже ранее фигурировали буквы E и F (такие ветви в графе, где из E допускается только один путь в Z, отметим синими пунктирными стрелками) — см. рис. на с. 10.

4) Остается подсчитать количество ветвей в этом графе, включающих в себя шесть и более вершин (т.е. когда запись последовательности имен проходимых вершин состоит из шести и более букв). Это ветви (пути): ABCDFZ, ABCDFEZ, ABCDEFZ, ABCDEFZ, ACDFEZ, ACDFEZ. Всего — шесть путей.

*Ответ:* 6.

3. Каждое из логических выражений F и G содержит шесть переменных. В таблицах истинности выражений F и G есть ровно десять одинаковых строк, причем ровно в пяти из них в столбце значений стоит 1.

Сколько строк таблицы истинности для выражения  $F \vee G$  содержат единицу в столбце значений?



*Решение<sup>1</sup>*

1) В каждом из логических выражений имеется шесть переменных, следовательно, для каждого выражения возможно  $2^6 = 64$  различных набора значений переменных.

2) По условию, в таблицах истинности выражений есть десять одинаковых строк (когда обе функции имеют значение 1 или значение 0). Значит, строк, в которых значения функций не совпадают, будет  $64 - 10 = 54$ .

3) Выражение  $F \vee G$  равно 1, если хотя бы одно из значений (F или G) равно единице, и равно нулю тогда и только тогда, когда оба значения (и F, и G) равны нулю. По условию, среди одинаковых строк в таблицах истинности F и G есть ровно пять строк, в которых значения равны 1. Значит, таких строк, где и для F, и для G значения равны нулю, будет  $10 - 5 = 5$  штук.

4) Итак, имеем:

- 54 строки, в которых значения F и G различны (одно равно 0, другое равно 1), т.е. значение выражения  $F \vee G$  равно 1;
- 5 строк, в которых значения F и G одинаковы и равны 1, т.е. значение выражения  $F \vee G$  также равно 1;
- 5 строк, в которых значения F и G одинаковы и равны 0, т.е. значение выражения  $F \vee G$  равно 0.

Условиям задачи соответствует, таким образом,  $54 + 5 = 59$  строк.

*Комментарий*

Вопрос мог быть поставлен и так: сколько строк содержат в столбце значений 1 для выражения  $F \wedge G$ ? При решении такой задачи ход рассуждений будет аналогичным, но на четвертом шаге надо учесть, что для логической операции И результат равен единице тогда и только тогда, когда оба значения (и F, и G) равны 1. Таких строк, очевидно, только пять.

Решение задач на определение количества строк таблицы истинности выражения  $F \vee G$ , содержащих нули, или количества строк таблицы истинности выражения  $F \wedge G$ , содержащих нули, оставим читателям для самостоятельного решения.

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.

В каталоге находится шесть файлов:

- maverick.map
- maverick.mp3
- taverna.mp4
- revolver.mp4
- vera.mp3
- zveri.mp3

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

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

*Решение*

Это — несколько усложненный вариант прежней задачи типа A4. Принцип решения — тот же: “примерка” каждой маски к именам файлов и определение количества файлов, соответствующих маске. Маска, для которой будет отобрано ровно три файла (или маски), — и есть ответ.

1) Маска **\*ver\*.mp\*** предполагает, что где-то в имени файла содержится набор символов **ver**, а расширение начинается с символов **mp**. Значит, такая маска отберет файлы **maverick.mp3**, **taverna.mp4**,

<sup>1</sup> Предложено Анастасией Ремизовой (11-й “А” класс, школа № 1360).

**revolver.mp4, vera.mp3, zveri.mp3** — всего пять файлов. Значит, эта маска нам не годится.

2) Маска **\*?ver?.mp?** отличается от предыдущей тем, что до и после символов **ver** обязательно должны стоять минимум по одному символу, а в расширении после символов **mp** может быть только один символ. Значит, такая маска отберет файлы **maverick.mp3, taverna.mp4** и **zveri.mp3** (а имена **revolver.mp4** и **vera.mp3** не удовлетворяют этой маске, так как в них группа символов **ver** располагается с краю имен). Всего — три файла. Значит, эта маска *подходит*.

3) Маска **\*?ver\*.mp?\*** — минимум один символ должен быть перед **ver**, а в расширении после **mp** должно быть не менее одного символа. Такая маска отберет файлы **maverick.mp3, taverna.mp4, revolver.mp4** и **zveri.mp3** — четыре файла. Данная маска не годится.

4) Маска **\*v\*r?.m?p\*** — не обращая внимания на маску имени, заметим: в расширении между **m** и **p** должен стоять хотя бы один какой-то символ. А у нас этому условию соответствует единственный файл **maverick.map**. Значит, эта маска тоже не годится.

5) Маска **???\*???.mp\*** — ей удовлетворяют файлы с любыми именами не менее чем из шести букв и с расширениями, начинающимися с **mp**. Значит, будут отобраны файлы **maverick.mp3, taverna.mp4** и **revolver.mp4** (у файлов **vera.mp3** и **zveri.mp3** имена слишком короткие, а расширение файла **maverick.map** не соответствует расширению в маске). Значит, данная маска тоже *подходит*.

6) Маска **???\*???.m\*** — отличается от предыдущей тем, что здесь расширение может быть любым, лишь бы оно начиналось с **m**. Тогда в дополнение к файлам **maverick.mp3, taverna.mp4** и **revolver.mp4** в подборку попадет и файл **maverick.map**. Всего — четыре файла. Значит, эта маска не годится.

7) Маска **\*a\*.a\*** — соответствует любым файлам, у которых и в имени, и в расширении есть хотя бы одна буква **a**. Этому условию соответствует только один файл — **maverick.map**. Значит, эта маска тоже не годится.

8) Маска **\*a\*.p\*** — здесь в имени требуется хотя бы одна буква **a**, а в расширении — хотя бы одна буква **p**. Этому условию соответствуют файлы: **maverick.map, maverick.mp3, taverna.mp4** и **vera.mp3**. В этой подборке четыре файла, значит, данная маска тоже не годится.

Итого нам подошло две маски — **\*?ver?.mp?** и **???\*???.mp\***.

*Ответ: 2.*

5. Автомат получает на вход четырехзначное число. По этому числу строится новое число по следующим правилам.

1. Складываются первая и вторая, а также третья и четвертая цифры исходного числа.

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

*Пример.* Исходное число: 6531. Суммы:  $6+5 = 11$ ,  $3+1 = 4$ . Результат: 411.

Укажите наибольшее число, в результате обработки которого автомат выдаст число 1113.

*Решение*

Подобные задачи в предыдущих вариантах ЕГЭ уже были. Но раньше нужно было выбрать подходящие числа из приведенных вариантов ответа, а теперь, наоборот, искать такое число.

1) Числа (по умолчанию) десятичные. Значит, суммы цифр могут быть в диапазоне от 1 ( $1+0$ ) до 18 ( $9+9$ ).

2) Автомат выдал число 1113. Значит, оно состоит из двух значений сумм, записанных по возрастанию: 11 и 13.

3) Число 11 может быть суммой цифр:  $2+9$ ,  $3+8$ ,  $4+7$  или  $5+6$ . Число 13 может быть суммой цифр:  $4+9$ ,  $5+8$ ,  $6+7$ . Значит, пары цифр в исходном числе могут быть такими: одна пара — 29, 92, 38, 83, 47, 74, 56, 65 и другая пара — 49, 94, 58, 85, 67, 76 (ведь соответствующие цифры могут меняться местами в соответствующих суммах).

4) Наибольшим является число, в которое входят наибольшие цифры, причем они по возможности располагаются слева направо по убыванию (поскольку чем старше разряд, тем больше “вес” цифры). Тогда из доступных нам “для конструирования” наборов пар цифр выберем две наибольшие (по одной из каждого набора): 92 и 94 и запишем их по убыванию: 9492.

*Ответ: 9492.*

6. В фрагменте базы данных представлены сведения о родственных отношениях. На основании приведенных данных определите, сколько прямых потомков (то есть детей и внуков) Кривич Л.П. упомянуто в таблице.

Таблица 1			Таблица 2	
ID	Фамилия И.О.	Пол	ID родителя	ID ребенка
2146	Кривич Л.П.	Ж	2146	2302
2155	Павленко А.К.	М	2146	3002
2431	Хитрук П.А.	М	2155	2302
2480	Кривич А.А.	М	2155	3002
2302	Павленко Е.А.	Ж	2302	2431
2500	Сокол Н.А.	Ж	2302	2511
3002	Павленко И.А.	М	2302	3193
2523	Павленко Т.Х.	Ж	3002	2586
2529	Хитрук А.П.	М	3002	2570
2570	Павленко П.И.	М	2523	2586
2586	Павленко Т.И.	Ж	2523	2570
2933	Симонян А.А.	Ж	2529	2431
2511	Сокол В.А.	Ж	2529	2511
3193	Биба С.А.	Ж	2529	3193
...	...	...	...	...

*Решение*

Данная задача решается так же, как привычные уже задачи типа А6, но здесь нужно, начиная от родителей, определять по базе данных сначала детей, по-

том внуков и т.д., подсчитывая их количество. (Аналогичным способом можно подсчитывать и количества других родственников.) Реально, как видно далее из решения, эта задача, наоборот, несколько упрощена по сравнению с прежними, — особенно теми, где нужно было искать племянников и племянниц.

1) В табл. 1 ищем Кривич Л.П. и определяем ее ID: 2146.

2) В правой таблице ищем все пары ID, в которых первое число (в левой колонке) равно 2146 (строки выделены зеленым цветом). Для них в правой колонке записаны ID 2302 и 3002. Это — дети Кривич Л.П.

3) В правой таблице ищем все пары ID, в которых первое число (в левой колонке) равно 2302 или 3002 (строки выделены, соответственно, голубым и желтым цветами). Это — внуки Кривич Л.П.

4) Подсчитываем количество закрашенных в табл. 2 строк: 7 (двое детей и пятеро внуков).

*Ответ:* 7.

7. Производилась двухканальная (стерео) звукозапись с частотой дискретизации 64 кГц и 24-битным разрешением. В результате был получен файл размером 480 Мбайт, сжатие данных не производилось. Определите приблизительно, сколько времени (в минутах) проводилась запись. В качестве ответа укажите ближайшее к времени записи целое число, кратное 10.

*Решение*

Это — вполне привычная задача типа А8. Только раньше мы, зная длительность записи, оценивали получаемый размер файла. А теперь требуется, наоборот, по заданному размеру файла оценить длительность. А значит, нужно будет составить уравнение.

1) Разрешение равно 24 битам.

2) Частота дискретизации равна 64 кГц — 64 000 измерений в секунду.

3) Длительность записи (в секундах) примем равной  $x$ .

4) Запись двухканальная — объем информации умножается на 2.

5) Результирующий объем файла равен 480 Мб, или  $480 \cdot 2^{23}$  бит.

(Не забываем, что все расчеты ведутся в битах и в секундах.)

6) Составляем уравнение:

$$24 \times 64\,000 \times x \times 2 = 480 \cdot 2^{23}.$$

Для удобства вычислений по возможности выделяем степени двойки:

$$3 \cdot 2^3 \times 125 \cdot 2^9 \times x \times 2 = 15 \cdot 2^5 \cdot 2^{23};$$

$$3 \cdot 5^3 \times x \times 2^{13} = 3 \cdot 5 \cdot 2^{28};$$

$$25 \times x = 2^{15};$$

$$x = 2^{15}/25 = 1310,72 \text{ (секунды)}.$$

7) Переводим  $x$  в минуты:

$$x \approx 1311 / 60 \approx 21,85 \text{ (минуты)}.$$

8) Округляем до ближайшего значения, кратного 10:  $x \approx 21,85 \approx 20$  (минут).

*Ответ:* 20.

8. 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 — 101, 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*

Задача на кодирование и применение условия Фано. Аналогичные задания были и в прежних вариантах ЕГЭ, но там требовалось лишь выбрать один из готовых вариантов ответа, а теперь нужно самим найти такой код.

Вспомним условие Фано:

**Неравномерный код может быть однозначно декодирован, если никакой из кодов не совпадает с началом (префиксом) какого-либо другого, более длинного кода.**

Таким образом, требуется найти кратчайший код, не совпадающий ни с одним имеющимся кодом, ни с началом одного из имеющихся кодов (Т — 101, О — 0, П — 100). Поиск решения ведем простым перебором от однобитовых кодов к кодам большей разрядности:

- код 0 — не годится (совпадает с кодом буквы О);

- код 1 — не годится (совпадает с началом кодов 100 и 101);

- код 10 — не годится (совпадает с началом кодов 100 и 101);

- код 11 — не совпадает с началом никакого из имеющихся кодов и ни с одним из имеющихся кодов. Следовательно, это и есть ответ.

*Ответ:* 11.

9. На числовой прямой даны два отрезка:

$$P = [43, 49] \text{ и } Q = [44, 53].$$

Укажите наибольшую возможную длину отрезка А, для которого формула

$$((x \in A) \rightarrow (x \in P)) \vee (x \in Q)$$

тождественно истинна, то есть принимает значение 1 при любом значении переменной  $x$ .

*Решение*

Уже привычная задача “на отрезки”, два способа решения которой описаны в статье авторов — “ЕГЭ: возможные особенности заданий 2014 года” (“Информатика”, 2014, № 5. С. 4–21). Впрочем, есть и еще один, более простой способ, который мы рассмотрим ниже.

1) Как обычно, заменяем высказывания логическими переменными:  $a = (x \in A)$ ,  $p = (x \in P)$ ,  $q = (x \in Q)$ . Тогда исходное логическое выражение принимает вид:

$$(a \rightarrow p) \vee q.$$

2) Избавимся от операции следования, используя закон де Моргана ( $x \rightarrow y = \neg x \vee y$ ):

$$(a \rightarrow p) \vee q = \neg a \vee p \vee q.$$

3) Требуется тождественное равенство этого выражения единице на всей числовой оси. При этом помним, что значения логических переменных рав-

ны 1 в пределах соответствующих отрезков и 0 — вне их (причем границы входят в отрезки).

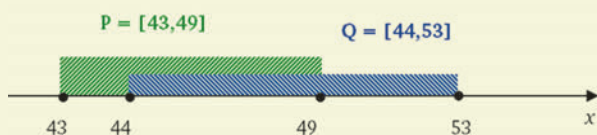
Анализируем полученное уравнение

$$\neg a \vee p \vee q = 1:$$

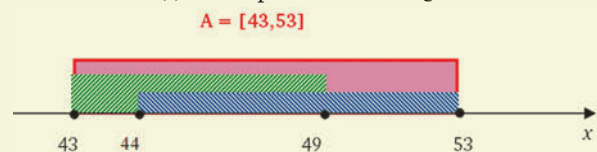
- переменные соединены операцией ИЛИ, следовательно, результат 1 будет автоматически обеспечен, если хотя бы одна из переменных —  $p$  или  $q$  — равна 1 (то есть везде, где числовая прямая “накрыта” хотя бы одним отрезком —  $P$  или  $Q$ );

- там, где нет ни отрезка  $P$ , ни отрезка  $Q$  (т.е. переменные  $p$  и  $q$  обе равны 0), для обеспечения истинности результата должна быть равна 1 переменная  $\neg a$ , или, соответственно, переменная  $a$  должна быть равна 0 (отрезка  $A$  не должно быть нигде, где нет ни отрезка  $P$ , ни отрезка  $Q$ ).

4) Рисуем числовую прямую и изображаем на ней отрезки  $P$  и  $Q$ :



5) Теперь совершенно очевидно, как расположить на этой числовой прямой отрезок  $A$ , чтобы он нигде “не высовывался” за пределы области, закрытой хотя бы одним отрезком  $P$  или  $Q$ :



6) А теперь — внимание! — нужно правильно вычислить длину полученного отрезка  $A$ , помня, что речь идет о прямой натуральных чисел. В данном случае это не имеет особого значения, но в других аналогичных задачах при выполнении операции отрицания границы отрезка  $A$  могут быть “выколоты” (могут не входить в отрезок), и тогда нужно при вычислении длины отрезка брать соседние натуральные граничные значения внутри отрезка. В нашем же случае конечные точки 43 и 53 входят в отрезок  $A$ .

Теперь вспомним, как мы вычисляем длину отрезка на обычной сантиметровой линейке, — например, между отметками 1 см и 3 см. Длина этого отрезка равна двум сантиметровым *интервалам* между указанными отметками на шкале линейки, т.е. вычисляется как разность значений. Точно так же нужно поступить и здесь, поэтому длина отрезка  $A$  равна  $53 - 43 = 10$ .

*Ответ:* 10.

**10.** При регистрации в компьютерной системе каждому пользователю выдается пароль, состоящий из восьми символов и содержащий только заглавные символы нижнего ряда латинской раскладки клавиатуры: Z, X, C, V, B, N, M. Каждый такой пароль 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 объем памяти (в байтах), отводимый этой программой для записи 25 паролей. В ответе запишите только число.

*Решение*

Это — все та же задача типа A11.

1) В алфавит входит семь символов (Z, X, C, V, B, N, M). При такой мощности алфавита для кодирования одного символа требуется три бита ( $2^3 = 8$ , а  $2^2 = 4$  — недостаточно).

2) Пароль состоит из восьми символов, значит, для их хранения потребуется  $8 \times 3 = 24$  бита.

3) Тогда для хранения одного пароля требуется  $24/8 = 3$  байта. (Если бы результат деления был не целым, потребовалось бы взять ближайшее большее целое количество байт.)

4) Для хранения 25 паролей тогда потребуется  $25 \times 3 = 75$  байт.

*Ответ:* 75.

**11.** В программе описан одномерный целочисленный массив с индексами от 0 до 10. Ниже представлен фрагмент программы, обрабатывающей данный массив<sup>2</sup>:

```
s := 0;
n := 10;
for i := 0 to n - 1 do begin
    s := s + A[i] - A[i + 1];
end;
```

В начале выполнения этого фрагмента в массиве находились трехзначные натуральные числа, ни одно из которых не делится на 20. Какое наибольшее значение может иметь переменная  $s$  после выполнения данной программы?

*Решение*

Задача похожа на привычное задание типа A12, за одним исключением: исходное содержимое массива неизвестно.

1) Анализируем алгоритм:

- длина массива равна 10; переменная  $s$  обнулена для подсчета суммы/разности;
- в цикле выполняется сложение прежнего значения  $s$  с разностью текущего элемента массива и последующего.

Такое вычисление значения  $s$  можно распisać в одну строку в виде:

$$s = (A[0] - A[1]) + (A[1] - A[2]) + (A[2] - A[3]) + (A[3] - A[4]) + (A[4] - A[5]) + (A[5] - A[6]) + (A[6] - A[7]) + (A[7] - A[8]) + (A[8] - A[9]) + (A[9] - A[10]).$$

$$\text{Раскроем скобки и перегруппируем слагаемые:} \\ s = A[0] + (A[1] - A[1]) + (A[2] - A[2]) + (A[3] - A[3]) + (A[4] - A[4]) + (A[5] - A[5]) + (A[6] - A[6]) + (A[7] - A[7]) + (A[8] - A[8]) - A[9] + (A[9] - A[9] + A[10]).$$

<sup>2</sup> В варианте диагностической работы, как и в вариантах ЕГЭ, в подобных случаях даются несколько листингов одной и той же программы, записанных на различных языках программирования: Бейсик, Паскаль, Си и “алгоритмический язык”. Мы же ради экономии места приводим только листинг на Паскале. — **Прим. авт.**

Как видим, большинство слагаемых сокращается, и получаем:

$$s = A[0] - A[10].$$

2) При этом требуется, чтобы итоговое значение  $s$  было максимальным. Когда это возможно, учитывая, что все числа — трехзначные и не кратные 20? Очевидно, уменьшаемое должно быть максимально возможным таким числом, а вычитаемое — минимально возможным. Всем этим условиям отвечают значения:  $A[0] = 999$ ,  $A[9] = 101$  (минимально возможное трехзначное число 100 непригодно, так как оно кратно 20). Тогда значение  $s$  (максимально возможное, требуемое в задаче) равно  $999 - 101 = 898$ .

Ответ: 898.

12. Система команд исполнителя РОБОТ, “живущего” в прямоугольном лабиринте на клетчатой плоскости, состоит из восьми 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 или вправо. Четыре других 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 команда1

ИНАЧЕ команда2

КОНЕЦ ЕСЛИ

выполняется команда1 (если условие истинно) или команда2 (если условие ложно).

В конструкциях ПОКА и ЕСЛИ условие может содержать команды проверки, а также слова И, ИЛИ, НЕ.

Если РОБОТ начнет движение в сторону находящейся рядом с ним стены, то он разрушится и программа прервется.

Сколько клеток лабиринта соответствуют требованию, что, начав движение в этой клетке и выполнив предложенную программу, РОБОТ уцелеет и остановится в закрашенной клетке (клетка F6)?

НАЧАЛО

ПОКА **снизу свободно ИЛИ справа свободно**

ЕСЛИ **снизу свободно**

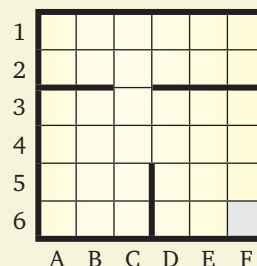
ТО **вниз**

ИНАЧЕ **вправо**

КОНЕЦ ЕСЛИ

КОНЕЦ ПОКА

КОНЕЦ

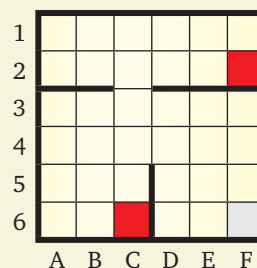


Решение

Привычная задача про РОБОТа (типа A13) теперь “обзавелась” новой особенностью алгоритма работы — полным ветвлением. Но принципы решения — все те же.

Анализируем предложенную программу РОБОТа.

• В начале очередного прохода цикла проверяется: есть ли стенки снизу и справа. Если хотя бы одно из этих направлений свободно, то выполняется очередной проход цикла. Значит, точка останова (клетка, в которой выполнение цикла корректно прекратится) имеет вид:  $\perp$ . Причем “правильной” является только одна такая клетка, а остальные две (позначим их красным цветом) — это “ловушки” для РОБОТа.

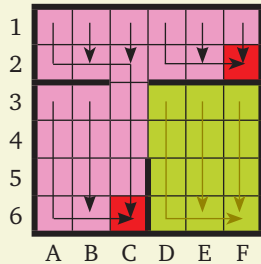


• Когда начинается очередной проход цикла, проверяется наличие стенки снизу. Если ее нет, то выполняется перемещение на одну клетку вниз. Если же стенка снизу есть, то выполняется перемещение на одну клетку вправо, причем *без проверки наличия стенки справа*. Следовательно, все клетки, в которых есть стенки снизу и есть стенки справа, можно было бы сразу пометить как непригодные (начав движение из них, РОБОТ разобьется). Но это не требуется: ведь такое условие записано в цикле ПОКА, который контролирует каждый отдельный шаг РОБОТа. (Вот если бы в цикле ПОКА было записано два или больше таких оператора ЕСЛИ, то для второго и т.д. такого оператора пришлось бы это исключение клеток сделать.)

• После выполнения одного такого хода проход цикла завершается, и вновь выполняется проверка его условия. Нетрудно понять, что в этом случае движение РОБОТа будет следующим: “идти вниз, пока это возможно (до стенки снизу), а наткнувшись на стенку снизу — пытаться уходить вправо, отыскивая проход вниз в этой стенке”.

Именно так мы и будем пытаться “трассировать” движение РОБОТа, “честно” выполняя предложенную программу. Увидев же, куда в итоге придет

РОБОТ, помечаем все пройденные клетки или розовым цветом (если попадаем в “ловушку”), или зеленым (если доходим до клетки F6). Сама же клетка F6 нам тоже подходит: если РОБОТ изначально находится в ней, то он просто никуда не пойдет и сразу же выполнит условие задачи.



Остается только подсчитать количество клеток, закрашенных зеленым: их 12.

Ответ: 12.

13. У исполнителя Аккорд две команды, которым присвоены номера:

1. отними 1

2. умножь на  $x$

( $x$  — неизвестное положительное число).

Выполняя первую из них, Аккорд отнимает от числа на экране 1, а выполняя вторую, умножает это число на  $x$ .

Программа для исполнителя Аккорд — это последовательность номеров команд.

Известно, что программа 12121 переводит число 4 в число 23.

Определите значение  $x$ .

Решение

Задача подобна прежним заданиям типа В1, но теперь нам нужно не составлять программу для исполнителя при известных командах, а, по сути, определить одну из команд при известной программе.

1) Попробуем расписать предложенную программу для исполнителя Аккорд, подставляя пока во второй команде значения  $x$  и заполнив левую колонку заданными исходным и конечным числами:

4	1	отними 1
	2	умножь на $x$
	1	отними 1
	2	умножь на $x$
	1	отними 1
23		

2) Очевидно, что для первой команды можно сразу записать результат ее выполнения. Точно так же можно определить число, которое было перед выполнением последней команды:

4	1	отними 1
3	2	умножь на $x$
	1	отними 1
	2	умножь на $x$
24	1	отними 1
23		

3) Вместо оставшихся числовых значений запишем соответствующие математические выражения и определим, какое математическое выражение должно давать известное нам предпоследнее число:

	4	1	отними 1
	3	2	умножь на $x$
	$3x$	1	отними 1
	$3x - 1$	2	умножь на $x$
	$x(3x - 1) = 24$	1	отними 1
	23		

4) Тем самым мы получили “ключевое” уравнение, позволяющее вычислить  $x$ :

$$x(3x - 1) = 24.$$

Решаем его как обычное квадратное уравнение:

$$3x^2 - x - 24 = 0;$$

$$D = 1 + 4 \cdot 3 \cdot 24 = 289 = 17^2;$$

$$x_{1,2} = (1 \pm 17)/(2 \cdot 3), \text{ тогда } x_1 = 18/6 = 3, \\ x_2 = -16/6 = -2^2/3.$$

Поскольку указано, что  $x$  — это положительное число, нам подходит только значение 3.

Ответ: 3.

14. Ниже приведен фрагмент программы. При каком наименьшем введенном числе  $a$  после выполнения программы значение переменной  $c$  будет равно 60?

```
readln(a);
b := 50;
a := a * 2 - b;
if a < b then
  c := b - 10
else
  c := b + 10;
```

Решение

В отличие от прежних заданий типа В2, здесь требуется не просто формально “протрассировать” заданную программу, а проанализировать алгоритм ее работы и понять, какими могут быть значения введенного числа  $a$ .

1) Цикла здесь нет, и изменение значения переменной  $b$  выполняется однократно (одним из операторов в ветвях then или else).

2) Изначально  $b = 50$ , а нам нужно, чтобы это значение стало равным 60. Следовательно, нам нужно для этого выполнить оператор  $c := b + 10$  в ветви else.

3) Чтобы выполнялась ветвь else, требуется, чтобы к моменту выполнения оператора if уже пересчитанное значение  $a$  ( $a := a * 2 - b$ ) было не меньше (т.е. больше или равно)  $b$ , т.е. числа 50.

4) Решаем полученное неравенство:

$$a \cdot 2 - 50 \geq 50; \Rightarrow 2a \geq 100; \Rightarrow a \geq 50.$$

5) Это — почти уже ответ. Нам же нужно определить, какое возможное значение  $a$  является наименьшим. Так как неравенство нестрогое, это значение  $a = 50$ .

Ответ: 50.



15. Сколько существует различных символьных последовательностей длины 5 в трехбуквенном алфавите {K, O, T}, которые содержат ровно две буквы O?

*Решение*

Задача подобна бывшим когда-то заданиям, в которых требовалось определить, какое слово стоит на указанном месте в списке слов, “синтезированных” из букв и отсортированных по порядку, например:

AAAAA  
AAAAO  
AAAOA  
AAAOO  
AAOAA  
...

Нетрудно вспомнить, что такие задачи мы решали, “переводя” буквы в цифры соответствующей системы счисления и работая с полученными числами. Здесь задача несколько иная, но принцип решения будет аналогичен.

1) Поскольку никаких слов изначально не задано, мы можем сами “назначить” цифры, соответствующие каждой букве. А поскольку букв в алфавите три, мы имеем дело с троичной системой счисления и выберем следующие соответствия букв и цифр:

O – 0, K – 1, T – 2.

Почему мы выбрали нуль именно для буквы O? А потому, что нам понадобится искать слова, содержащие ровно две буквы O, а с нулями это, возможно, окажется проще.

2) Итак, речь идет об определении количества пятиразрядных троичных чисел (причем незначащие нули слева важны!), содержащих ровно два нуля.

3) Первый нуль может быть в одной из пяти позиций — получаем пять вариантов.

Как в каждом из этих вариантов может располагаться второй нуль? Посмотрим:

- когда первый нуль находится в позиции “1”, второй нуль может располагаться в одной из четырех оставшихся позиций — “2”, “3”, “4”, “5”;

- когда первый нуль находится в позиции “2”, второй нуль может располагаться в одной из трех оставшихся позиций правее — “3”, “4” и “5” (ведь ситуацию, когда нули располагаются в позициях “1” и “2”, мы уже рассмотрели перед этим);

- когда первый нуль находится в позиции “3”, второй нуль может располагаться в одной из двух оставшихся позиций правее — “4” и “5” (почему, — рассмотрено выше);

- когда первый нуль находится в позиции “4”, второй нуль может располагаться только в одной позиции правее — “5”.

Итого получаем  $4 + 3 + 2 + 1 = 10$  вариантов размещения в пятиразрядном числе двух нулей.

Возможны и другие рассуждения, приводящие к тому же результату.

Условно обозначим наши нули разными цветами: например, первый — синим, а второй — красным.

Тогда синий нуль можно разместить в пяти рядах — получаем пять вариантов. И в каждом из этих пяти вариантов красный нуль можно разместить в любой из оставшихся четырех позиций — т.е. по четырем вариантам. Значит, всего получаем  $5 \cdot 4 = 20$  вариантов размещения наших разноцветных нулей.

Но теперь вспомним, что на самом деле оба нуля (синий и красный) — совершенно равноправны. А значит, пары “синий — красный” и “красный — синий” (если читать число слева направо) — это одни и те же пары. Следовательно, каждая такая пара посчитана дважды, и всего вариантов размещения в пятиразрядном числе двух одинаковых нулей будет 10.

Итак, существует десять вариантов размещения в числе двух нулей. В каждом из этих десяти вариантов остается три цифры, которые могут быть равны или 1, или 2.

Сколько может быть таких неповторяющихся комбинаций? Очевидно, столько, сколько может быть различных трехразрядных чисел в системе счисления, состоящей из двух цифр (т.е. двоичной). Значит, для каждого из десяти ранее найденных вариантов получается  $2^3 = 8$  “подвариантов”.

Тогда общее число троичных чисел, в которых из пяти цифр ровно две — нулевые, будет равно  $8 \cdot 10 = 80$ .

*Ответ:* 80.

16. Ниже приведена программа. При каком наибольшем введенном числе  $d$  после выполнения программы будет напечатано 55?

```
var n, s, d: integer;
begin
  readln(d);
  n := 0;
  s := 0;
  while s <= 365 do
  begin
    s := s + d;
    n := n + 5;
  end;
  write(n);
end.
```

*Решение*

Опять же, задача по виду подобна прежним заданиям В5, но ранее там требовалось определить конечное значение вычисляемой переменной, а теперь — наоборот, искать по заданному конечному значению исходное.

1) Анализируем алгоритм:

- в цикле ПОКА задано условие выполнения цикла — до тех пор, пока значение  $s$  не превышает 365;
- внутри цикла значение переменной  $s$  увеличивается на заданное значение  $d$ , а значение переменной  $n$  увеличивается на 5;
- по завершении цикла на экран выводится значение переменной  $n$ .

2) Тогда, согласно условию, значение  $n$  равно 55. Учитывая, что изначально переменная  $n$

обнуляется, это означает, что цикл выполнен 55/5 = 11 раз.

3) Учтявая, что переменная  $s$  изначально тоже обнуляется, получаем, что за эти 11 проходов цикла переменная  $s$  увеличивается на  $11d$  и становится больше 365 (условие прекращения работы цикла). Тогда получаем неравенство:

$$11d > 365,$$

откуда  $d > 365/11$ , т.е.  $d > 33^{2/11}$ , или, учитывая, что числа — целые,  $d > 33$ .

4) Чтобы теперь оценить верхнюю границу интервала возможных значений  $d$ , надо учесть, что при слишком большом значении  $d$  цикл мог бы прерваться еще после десяти проходов (а то и раньше). Значит, нам нужно определить значение  $d$ , на единицу меньшее, чем минимальное значение  $d$ , обеспечивающее выполнение десяти проходов цикла. В этом случае неравенство будет таким:

$$10d > 365,$$

тогда  $d > 365/10$ , или  $d > 36,5$ .

5) Рассуждая аналогично п. 3, можно заключить, что минимальное значение  $d$ , при котором цикл завершится через десять проходов, равно 37 (ближайшее целое число, большее 36,5). Нам же требуется значение  $d$ , на единицу меньшее. Это — число 36.

Таким образом, диапазон возможных значений  $d$ , обеспечивающих выполнение 11 проходов цикла, составляет [34 .. 36], а искомое максимальное значение  $d$ , при котором будет выведено число 55, равно 36.

*Ответ:* 36.

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

$$F(n) = 1 \text{ при } n \leq 2;$$

$$F(n) = F(n-2) \times (n+2) \text{ при } n > 1.$$

Чему равно значение функции  $F(8)$ ?

*Решение*

Это — задача на рекурсию, практически неизменно повторяющая задание типа В6 из демоварианта 2014 года.

Решение заключается в аккуратном расписывании всех значений  $F(n)$  для возрастающих значений  $n$  от 1 до требуемого значения 8, при этом для вычисления очередного значения используются ранее вычисленные предыдущие значения  $F(n)$ :

$$F(0) = 1;$$

$$F(1) = 1;$$

$$F(2) = F(0) \cdot (2+2) = 4;$$

$$F(3) = F(1) \cdot (3+2) = 5;$$

$$F(4) = F(2) \cdot (4+2) = 24;$$

$$F(5) = F(3) \cdot (5+2) = 35;$$

$$F(6) = F(4) \cdot (6+2) = 192;$$

$$F(7) = F(5) \cdot (7+2) = 315;$$

$$F(8) = F(6) \cdot (8+2) = 1920.$$

Последнее вычисленное значение — и есть искомый ответ. Заметим, что  $F(7)$  для получения ответа на вопрос задачи можно и не вычислять.

*Ответ:* 1920.

18. Известно, что для целого числа  $x$  выполнено:

$$33_{4+x} - 33_4 = 33_{10}.$$

Определите значение  $x$ . Ответ запишите в десятичной системе счисления.

*Решение*

По сути, это — несколько усложненное (по формулировке условия) и вместе с тем — немного упрощенное (по степени сложности решения) © задание типа В7 на поиск основания системы счисления.

1) Запишем “каноническое” преобразование обоих недесятичных чисел в десятичную систему счисления (согласно формуле вида  $a_1 \cdot n^1 + a_0 \cdot n^0$ ):

$$\bullet 33_{4+x} = 3 \cdot (4+x)^1 + 3 \cdot (4+x)^0 = \\ = 3 \cdot (4+x) + 3 = 12 + 3x + 3 = 15 + 3x;$$

$$\bullet 33_4 = 3 \cdot 4^1 + 3 \cdot 4^0 = 3 \cdot 4 + 3 = 15.$$

2) Теперь, когда все числа переведены в десятичную систему счисления, можно записать уравнение в виде:

$$15 + 3x - 15 = 33,$$

$$\text{откуда } 3x = 33, \text{ тогда } x = 11.$$

*Ответ:* 11.

19. Записан алгоритм. Получив на вход число  $x$ , этот алгоритм печатает два числа:  $a$  и  $b$ . Сколько существует таких чисел  $x$ , при вводе которых алгоритм печатает сначала 2, а потом 15?

```
var x, a, b: integer;
begin
  readln(x);
  a := 0; b := 0;
  while x > 0 do
  begin
    a := a + 1;
    b := b + (x mod 10);
    x := x div 10;
  end;
  writeln(a); writeln(b);
end.
```

*Решение*

Нетрудно увидеть, что эта задача — почти аналог прежнего задания типа В8. Но если ранее надо было определить одно входное число (например, наименьшее из возможных), то теперь у нас спрашивают их количество.

1) Как всегда, начинаем с анализа алгоритма:

• цикл ПОКА выполняется до тех пор, пока  $x$  не станет нулем;

• в цикле значение переменной  $a$  каждый раз увеличивается на 1, следовательно,  $a$  — это счетчик количества проходов цикла;

• переменная  $b$  служит для накапливания суммы значений  $x \bmod 10$ , а затем переменная  $x$  (исходное число) переприсваивается на значение  $x \div 10$ . Такой алгоритм должен знать каждый школьник: это — разбор в цикле исходного числа  $x$  на отдельные цифры (справа налево), а в переменной  $b$ , таким образом, вычисляется сумма цифр исходного числа;

• после цикла выводится сначала значение переменной  $a$ , а затем — переменной  $b$ .

2) По условию, в ходе выполнения алгоритма были выведены числа 2 и 15, — значит,  $a = 2$ ,  $a b = 15$ . Следовательно, цикл выполнился два раза ( $a$  значит, в исходном числе  $x$  было две цифры), а сумма цифр этого числа равна 15.

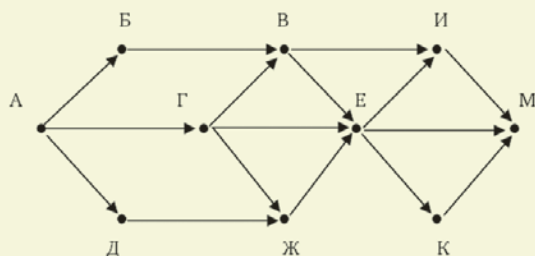
3) Для ответа на вопрос задачи нужно определить, сколько существует двузначных натуральных<sup>3</sup> чисел, сумма цифр которых равна 15. Их нетрудно найти простым перебором (учитывая и то, что суммы цифр могут быть и “симметричными”). Для этого по очереди предполагаем первую цифру равной 1, 2, 3 и т.д. и определяем, чему может быть равна вторая цифра. Получаем:

69, 96, 78, 87.

Всего — четыре числа.

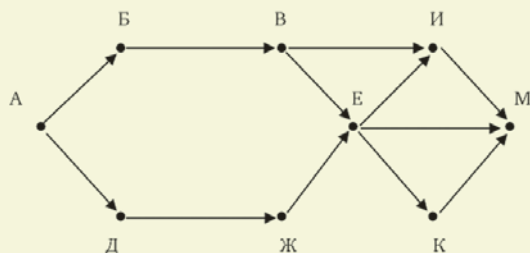
Ответ: 4.

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

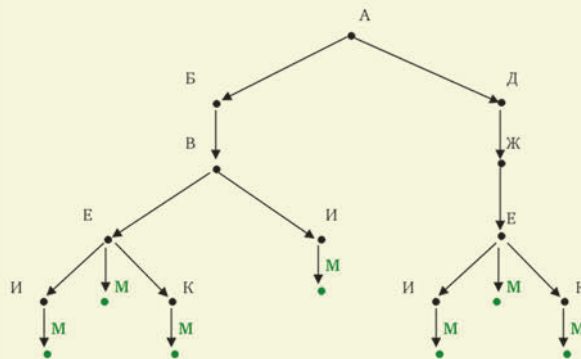


Решение

Такие задачи (тип В9) мы решать уже умеем (см. статью: Богомолова О.Б., Усенков Д.Ю. ЕГЭ: возможные особенности заданий 2014 года // “Информатика”, 2014, № 5. С. 4–21). Ведь применяемый для этого обычно способ с построением дерева путей универсален: он дает все возможные пути из одного города в другой, а уж найти среди них проходящие через “запретный” для нас город Г и исключить их из подсчета легко. Но мы сделаем еще проще (и тем самым существенно облегчим себе решение): раз через город Г идти нельзя, его можно сразу исключить из схемы!



Теперь и строить дерево путей гораздо проще:



Остается подсчитать количество конечных точек, соответствующих городу М (выделены в дереве зеленым цветом). Их — 7.

Ответ: 7.

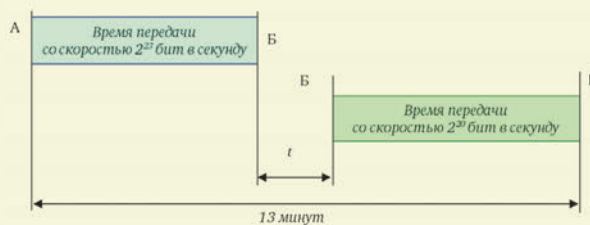
21. Данные объемом 80 Мбайт передаются из пункта А в пункт В по каналу связи, обеспечивающему скорость передачи данных  $2^{23}$  бит в секунду, а затем из пункта В в пункт В по каналу связи, обеспечивающему скорость передачи данных  $2^{20}$  бит в секунду. От начала передачи данных из пункта А до их полного получения в пункте В прошло 13 минут.

Через какое время в секундах началась передача данных в пункте В, т.е. каково время между началом передачи данных из пункта А и началом передачи данных в пункт В? В ответе укажите только число, слово “секунд” или букву “с” добавлять не нужно.

Решение

Ну а это — хорошо нам знакомая задача на передачу данных (см. статью: Богомолова О.Б., Усенков Д.Ю. “Задача о передаче”: решение задачи ЕГЭ при помощи сетевых диаграмм Ганта // “Информатика”, 2011, № 7. С. 7–8), которая предлагалась на ЕГЭ три-четыре года назад. Только там нам нужно было искать общее время передачи данных из А в В, а теперь придется составлять уравнение с одним неизвестным.

1) Построим диаграмму Ганта для этой задачи (как это делается, читатели могут узнать в вышеуказанной статье, а здесь мы вдаваться в подробности не будем):



При этом следует учитывать, что процесс передачи данных из В в В по условию задачи начинается уже после того, как завершится передача данных из А в В.

2) Составляем по этой схеме уравнение ( $t$  — это время задержки между окончанием приема данных из А и началом их передачи в В). Для этого вычисляем время передачи данных из А в В и из В в В:

<sup>3</sup> То, что вводимое число  $x$  — натуральное, в условии не оговаривается, но предполагается, так как при  $x \leq 0$  заданный в программе цикл работать не будет вовсе. — Прим. авт.

- из А в Б:  $80 \text{ (Мбайт)} / 2^{23} \text{ (бит в секунду)} = 5 \cdot 2^{27} \text{ (бит)} / 2^{23} \text{ (бит в секунду)} = 5 \cdot 2^4 = 80 \text{ (с)}$ ;
- из Б в В:  $80 \text{ (Мбайт)} / 2^{20} \text{ (бит в секунду)} = 5 \cdot 2^{27} \text{ (бит)} / 2^{20} \text{ (бит в секунду)} = 5 \cdot 2^7 = 640 \text{ (с)}$ .

Записываем уравнение:

$$80 + t + 640 = 13 \text{ (минут)} = 13 \cdot 60 \text{ (с)}, \text{ т.е. } 720 + t = 780, \text{ откуда } t = 60.$$

3) Чтобы найти время от начала передачи данных из А в Б до начала их передачи из Б в В, нужно к найденному значению  $t$  прибавить время передачи данных из А в Б:

$$80 + 60 = 140 \text{ (с)}.$$

Ответ: 140.

22. 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 четырех байтов, причем каждый байт записывается в виде десятичного числа. При этом в маске сначала (в старших разрядах) стоят единицы, а затем с некоторого места — 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 подсети равен 231.32.240.0.

Для узла с IP-адресом 224.128.112.142 адрес сети равен 224.128.64.0. Чему равен третий слева байт маски? Ответ запишите в виде десятичного числа.

Решение

Такие задачи на определение адреса сети по заданному IP-адресу и маске уже всем знакомы (тип В11). Но здесь нам нужно искать не адрес сети, а, наоборот, маску по заданным IP-адресу и адресу сети. Причем поскольку для байтов, которые в указанных адресах совпадают, байт маски очевидно равен 255, а там, где в адресе сети нулевой байт, байт маски тоже, очевидно, равен нулю, экзаменаторов интересует именно “ключевой” байт маски — третий слева. Так что задача для нас даже немного упростилась.

1) Преобразуем заданные третьи слева байты исходного IP-адреса и адреса сети в двоичные числа:

- $112_{10} = 1110000_2$ ;
- $64_{10} = 1000000_2$ .

2) Для получения из исходного IP-адреса заданного адреса сети требовалась поразрядная конъюнкция (для каждого бита) IP-адреса и маски, т.е.:

$$\begin{array}{r} 1110000 \\ \& \text{ *****} \\ 1000000 \end{array}$$

Биты маски нам не известны. Но их очень легко определить:

- там, где значения битов IP-адреса и адреса сети оба равны 1, бит маски должен быть тоже равен 1;
- там, где бит IP-адреса равен 1, а бит адреса сети равен 0, бит маски должен быть равен 0.

Единственная неоднозначность — там, где биты IP-адреса и адреса сети оба равны нулю: тут бит маски может быть равен как 0, так и 1. Но устранить эту неоднозначность нетрудно, зная, что в маске при ее просмотре слева направо может быть только один переход от единичных битов к нулевым, т.е., например, маска 11000110 недопустима. А мы видим, что уже со второго слева бита в маске должны начаться нули. Значит, и во всех “спорных” битах маски правее этого места тоже должны быть только нули.

3) Теперь нетрудно восстановить вышеприведенный “ребус” с конъюнкцией двоичных чисел:

$$\begin{array}{r} 1110000 \\ \& \underline{1000000} \\ 1000000 \end{array}$$

4) Остается перевести найденную двоичную маску в десятичное число. Для этого даже не понадобится никаких вычислений — полученная маска полностью совпала с двоичной записью байта адреса сети ( $64_{10}$ ), значит, и искомым байт маски тоже равен 64.

Ответ: 64.

23. 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)
США   Япония   Китай	450
Япония   Китай	260
(США & Япония)   (США & Китай)	100

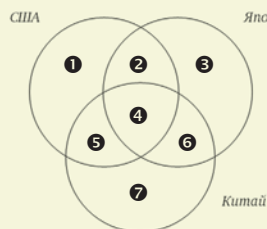
Какое количество страниц (в тысячах) будет найдено по запросу США?

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

Решение

Еще одна хорошо знакомая нам задача (тип В12). Но один из запросов, приведенных в таблице, стал сложнее. Впрочем, принцип решения остается прежним (см. статью: Богомолова О.Б., Усенков Д.Ю. «В поисках поиска»: задачи ЕГЭ, посвященные поиску информации на сайтах» // “Информатика”, 2011, № 11. С. 17–23).

1) Рисуем примерную диаграмму Эйлера – Венна и нумеруем области на ней:



2) Записываем уравнения с “номера-переменными” по заданным в условии запросам. При этом особое внимание обратим на сложный запрос в третьей строке таблицы.

США   Япония   Китай	$\textcircled{1} + \textcircled{2} + \textcircled{3} + \textcircled{4} + \textcircled{5} + \textcircled{6} + \textcircled{7} = 450$
Япония   Китай	$\textcircled{2} + \textcircled{3} + \textcircled{4} + \textcircled{5} + \textcircled{6} + \textcircled{7} = 260$
(США & Япония)   (США & Китай) = США & (Япония   Китай) США	$\textcircled{2} + \textcircled{4} + \textcircled{5} = 100$
	$\textcircled{1} + \textcircled{2} + \textcircled{4} + \textcircled{5} = ?$

3) Теперь главное — найти, как удобнее всего решить эту систему уравнений. Но это неожиданно легко.

Сразу очевидно, что из уравнений:

$$\textcircled{1} + \textcircled{2} + \textcircled{3} + \textcircled{4} + \textcircled{5} + \textcircled{6} + \textcircled{7} = 450$$

и

$$\textcircled{2} + \textcircled{3} + \textcircled{4} + \textcircled{5} + \textcircled{6} + \textcircled{7} = 260$$

можно найти значение  $\textcircled{1}$ , равное  $450 - 260 = 190$ .

Тогда, зная, что  $\textcircled{2} + \textcircled{4} + \textcircled{5} = 100$ , сразу можно вычислить, что  $\textcircled{1} + \textcircled{2} + \textcircled{4} + \textcircled{5} = 190 + 100 = 290$ .

Ответ: 290.

24. Определите, какое значение  $H$  нужно ввести, чтобы число, напечатанное в результате выполнения следующего алгоритма, было наименьшим.

```

var a, b, t, M, R, H: integer;
Function F(H, x: integer): integer;
begin
  F := 11*(x - H)*(x - H) + 13;
end;
begin
  readln(H);
  a := -10; b := 50;
  M := a; R := F(H, a);
  for t := a to b do begin
    if (F(H, t) > R) then begin
      M := t;
      R := F(H, t);
    end;
  end;
  write(R);
end.

```

Решение

А это — несколько усложненное задание типа В14 (с параболой). Такие задачи уже встречались ранее, но тогда требовалось для квадратичной функции одной переменной  $F(x)$  найти, какое число выводится в результате. А теперь у нас — функция с двумя переменными  $F(H, x)$ , которая представляет собой выражение с параметром  $H$ , и искать надо значение этого параметра, такое, чтобы выводимое значение было наименьшим.

Прежде всего заметим, что работа в языке программирования (в данном случае в Паскале, но это верно и для других языков — по крайней мере для большинства из них) с функциями с одной и с двумя переменными полностью идентична: в обоих случаях в функцию передаются значения переменных,

а результат возвращается через саму функцию  $F()$ . Поэтому сосредоточимся на собственно решении.

1) Анализируем алгоритм.

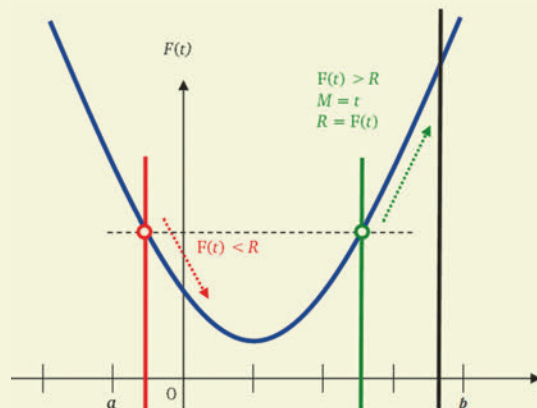
- В цикле идет перебор значений  $t$  от заданных ранее  $a = -10$  до  $b = 50$ . При этом в переменной  $M$  запоминается сначала начальное значение  $a$ , а потом (при срабатывании условного оператора) — текущее значение  $t$ , а в переменной  $R$  хранится предыдущее значение функции  $F(H, x)$ .

- Если с увеличением значения  $t$  получаемое значение  $F()$  увеличивается (и в результате оказывается больше предыдущего значения  $F(t)$ , запомненного в переменной  $R$ ), текущее значение  $t$  заносится в переменную  $M$ , а значение функции в этой точке запоминается в  $R$ . То есть ищется максимум функции  $F()$  на интервале  $[a; b]$ .

- До каких пор это будет продолжаться? Посмотрим на выражение, записанное в составе функции  $F()$ . Выполним в нем перемножение скобок. Получим:

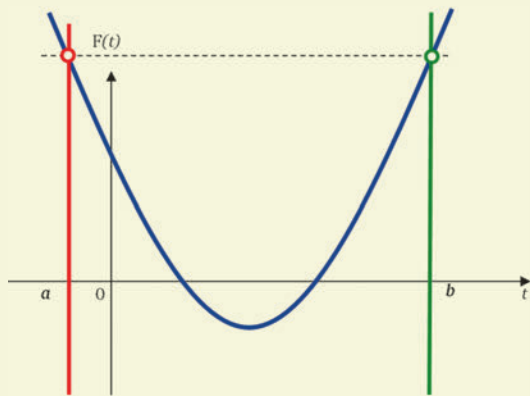
$$11*(x-H)*(x-H) + 13 = 11x^2 - 22xH + (11H^2 + 13).$$

Графиком этой функции от  $x$  является парабола. Поскольку перед  $x^2$  стоит знак “плюс”, ветви этой параболы направлены вверх. И мы в своих вычислениях “движемся” по левой ветви этой параболы сверху вниз. Значит, условный оператор в цикле не будет работать до тех пор, пока мы не пройдем вершину параболы (минимум на ее графике) и не поднимемся по правой ветви настолько, что значение выражения  $11t^2 - 22tH + (11H^2 + 13)$  (замена переменной  $x$  на  $t$  происходит при переходе из основной программы в функцию) в некоторой точке  $t$  превысит его значение в исходной точке  $a$ . А после этого, пока мы “движемся” по правой ветви параболы вверх, у нас значение  $R$  будет расти вплоть до окончания цикла по достижении точки  $b$  (см. примерный рисунок). Следовательно, конечное значение  $R$  будет равно значению выражения  $F(H, b) = 11b^2 - 22bH + (11H^2 + 13)$  либо, если расположить параболу так, что указанное значение меньше чем исходное ( $F(H, a) = 11a^2 - 22aH + (11H^2 + 13)$ ), значение  $R$  будет равно этому исходному.



2) Как нужно расположить параболу, чтобы получить минимально возможное значение  $R$ ? И каким оно будет? Очевидно, что если ветвь then нач-

нет работать, то это приведет к увеличению значения  $R$ . Следовательно, наименьшее значение  $R$  (которое и выводится на экран в результате работы программы) будет достигаться, если ветвь `then` так ни разу и не сработает (и будет равным значению в исходной точке  $a$ ). Для этого требуется, чтобы в точках  $a = -10$  и  $b = 50$  значения выражения, записанного в функции, были равными. Отсюда очевидно, что  $H = (-10 + 50) / 2 = 20$ .



Ответ: 20.

На этом перечень задач, входивших в пробную диагностику, заканчивается. Задачи группы С в предлагаемую подборку заданий не входили. (Впрочем, это вполне понятно — ребятам хватило и приведенных в данной статье усложненных задачек группы В, тем более что по нормативам на их решение отводилось всего 90 минут — т.е. менее четырех минут на одно задание!) Но что может появиться в группе С, если “экстраполировать” обнаруженные усложнения и на нее, — пока даже страшно и думать. Остается лишь надеяться, что, получив результаты (которые наверняка будут крайне низкими во всех школах, так как к подобным задачам большинство школьников не готовы) и ужаснувшись им, организаторы ЕГЭ и составители заданий все же поймут, что нужно немного снизить чересчур завышенные требования...



## СПРАВОЧНИК ЕГЭ\*

**IP-адрес** — уникальный (не повторяющийся) цифровой индивидуальный номер компьютера в сети. IP-адрес может быть закреплен за данным компьютером постоянно (**выделенный IP-адрес**) или выдаваться компьютеру временно из некоторого имеющегося набора только на время данного сеанса работы в сети.

В стандарте IP v.4 (рассматриваемом в задачах ЕГЭ) IP-адрес представляет собой запись из четырех разделенных точками чисел, каждое из которых соответствует одному байту и имеет значение от 0 до 255. Пример:

**168.154.0.177**

Некоторые IP-адреса являются служебными и используются для специальных целей:

- адрес сети, в котором один или несколько крайних справа байтов равны нулю (примеры: **168.154.15.0**, **168.154.0.0**);
- широковещательный адрес, в котором один или несколько крайних справа байтов равны 255 (примеры: **168.154.15.255**, **168.154.255.255**).

Если компьютер находится в составе локальной сети, которая, в свою очередь, подключена к сети Интернет, то возникает задача — определить IP-адрес всей локальной сети в целом (например, чтобы разослать пакет информации на *все* компьютеры данной локальной сети) и адрес конкретного компьютера внутри этой локальной сети.

Адрес сети и адрес компьютера в этой сети — это две условно выделяемые части единого IP-адреса данного компьютера, под которым он числится в Интернете:



\* По материалам пособия: Богомолова О.Б. Информатика: ЕГЭ за 30 дней: экспресс-репетитор. М.: АСТ, Астрель, 2014.