

издательский дом «Первое сентября»

ИНФОРМАТИКА А

**Как решать
принципиально новые задачи
ЕГЭ-2015**

Богомолова О.Б.
Усенков Д.Ю.

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) Чем левее в двоичном числе единица, тем бóльшая степень двойки ей соответствует. Максимальное значение двоичного числа получится, если все единицы будут «собраны» в старших разрядах.
- 3) В записи числа 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 в десятичной форме: $11000_2 = 2^4 + 2^3 = 16 + 8 = 24_{10}$.

Ответ: 24.

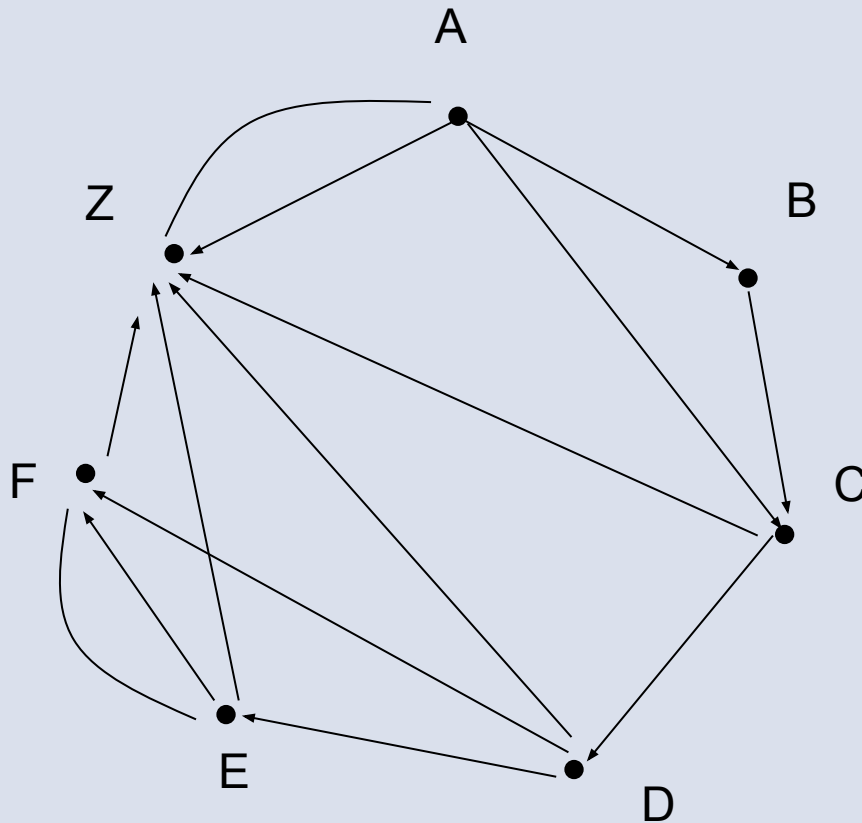
Между населенными пунктами **A, B, C, D, E, F, Z** построены дороги с односторонним движением. В таблице указана протяженность каждой дороги. Отсутствие числа в таблице означает, что прямой дороги между пунктами нет. Например, из **A** в **B** есть дорога длиной 4 км, а из **B** в **A** дороги нет.

	A	B	C	D	E	F	Z
A		4	6				30
B			3				
C				11			27
D					4	7	10
E						4	8
F					5		2
Z							

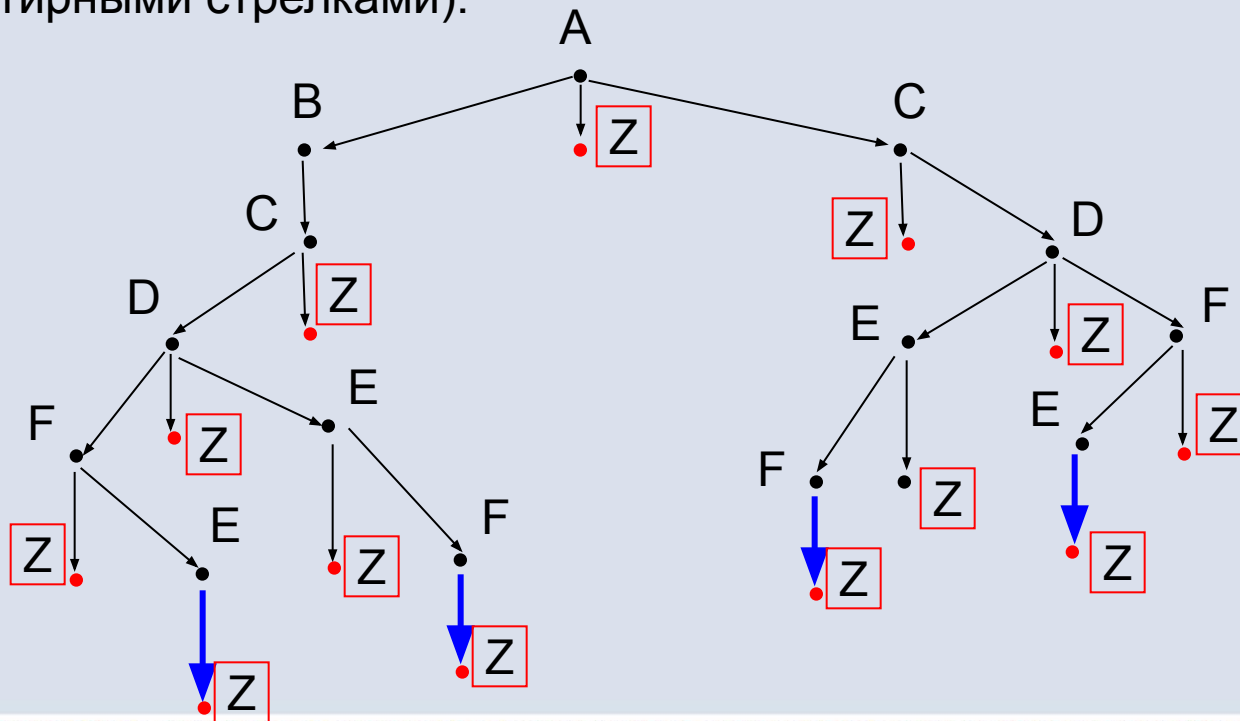
Сколько существует таких маршрутов из **A** в **Z**, которые проходят через 6 и более населенных пунктов? Пункты **A** и **Z** при подсчете учитывать. Два раза проходить через один пункт нельзя.

Решение

Строим по таблице **ориентированный граф**:



Рисуем дерево всех возможных путей из **A** в **Z**, при этом конечные точки маршрута (**Z**) выделяем цветом и рамкой. Не забываем о запрете на повторное посещение вершин: обратный путь **FE** нельзя рисовать в тех ветвях дерева, где уже где-то ранее фигурировали буквы **E** и **F** (такие ветви в графе, где из **E** допускается только один путь в **Z**, отметим синими пунктирными стрелками):



Остается подсчитать количество ветвей в этом графе, включающих в себя 6 и более вершин (т.е. когда запись последовательности имен проходимых вершин состоит из 6 и более букв).

Это ветви (пути):

ABCDFZ,

ABCDFEZ,

ABCDEZ,

ABCDEFZ,

ACDFEZ,

ACDEFZ.

Всего – 6 путей.

Ответ: 6.

Каждое из логических выражений **F** и **G** содержит 6 переменных.
В таблицах истинности выражений **F** и **G** есть ровно 10 одинаковых строк,
причем ровно в 5 из них в столбце значений стоит 1.

Сколько строк таблицы истинности для выражения **F** \vee **G** содержит 1
в столбце значений?

Решение

- 1) В каждом из логических выражений имеется 6 переменных, следовательно, для каждого выражения возможно $2^6 = 64$ различных набора значений переменных.
- 2) В таблицах истинности выражений есть 10 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. Среди одинаковых строк в таблицах истинности **F** и **G** есть ровно 5 строк, в которых значения равны 1. Значит, таких строк, где и для **F**, и для **G** значения равны нулю, будет $10 - 5 = 5$ штук.

Решение

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.

Ответ: 59.

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

- Символ «?» (вопросительный знак) означает ровно один произвольный символ.
- Символ «*» (звёздочка) означает любую последовательность символов произвольной длины, в том числе пустую последовательность.

В каталоге находится 6 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*

6 файлов:

maverick.map
revolver.mp4

maverick.mp3
vera.mp3

taverna.mp4
zveri.mp3

Решение

- 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**, **revolver.mp4**, **vera.mp3**, **zveri.mp3** – всего 5 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 – 3 файла. Значит, эта маска **подходит**.

6 файлов:

maverick.map

revolver.mp4

maverick.mp3

vera.mp3

taverna.mp4

zveri.mp3

Решение

- 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** – 4 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**.

6 файлов:

maverick.map
revolver.mp4

maverick.mp3
vera.mp3

taverna.mp4
zveri.mp3

Решение

- 5) Маска **???*???.mp*** – ей удовлетворяют файлы с любыми именами не менее чем из 6 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 – 4 файла. Значит, эта маска **не годится**.

6 файлов:

maverick.map
revolver.mp4

maverick.mp3
vera.mp3

taverna.mp4
zveri.mp3

Решение

- 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 4 файла, значит, данная маска тоже **не годится**.

Итого нам подошло две маски – ***?ver?.mp?** и **???*???.mp***.

Ответ: 2.

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

- 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**
(соответствующие цифры могут меняться местами в суммах).
- 4) Наибольшим является число, в которое входят наибольшие цифры, причем они по возможности располагаются слева направо по убыванию (чем старше разряд, тем больше «вес» цифры). Тогда из доступных наборов пар цифр выберем две наибольшие (по одной из каждого набора): **92** и **94** и запишем их по убыванию: **9492**.

Ответ: 9492.

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. упомянуты в таблице.

ID	Фамилия И.О.	Пол
2146	Кривич Л.П.	Ж
2155	Павленко А.К.	М
2431	Хитрук П.А.	М
2480	Кривич А.А.	М
2302	Павленко Е.А.	Ж
2500	Сокол Н.А.	Ж
3002	Павленко И.А.	М
2523	Павленко Т.Х.	Ж
2529	Хитрук А.П.	М
2570	Павленко П.И.	М
2586	Павленко Т.И.	Ж
2933	Симомян А.А.	Ж
2511	Сокол В.А.	Ж
2102	Борисов С.А.	Ж

ID родителя	ID ребенка
2146	2302
2146	3002
2155	2302
2155	3002
2302	2431
2302	2511
2302	3193
3002	2586
3002	2570
2523	2586
2523	2570
2529	2431
2529	2511
2529	2102

Решение

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

ID	Фамилия И.О.	Пол
2146	Кривич Л.П.	Ж
2155	Павленко А.К.	М
2431	Хитрук П.А.	М
2480	Кривич А.А.	М
2302	Павленко Е.А.	Ж
2500	Сокол Н.А.	Ж
3002	Павленко И.А.	М
2523	Павленко Т.Х.	Ж
2529	Хитрук А.П.	М
2570	Павленко П.И.	М

ID родителя	ID ребенка
2146	2302
2146	3002
2155	2302
2155	3002
2302	2431
2302	2511
2302	3193
3002	2586
3002	2570
2523	2586

Решение

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

ID	Фамилия И.О.	Пол
2146	Кривич Л.П.	Ж
2155	Павленко А.К.	М
2431	Хитрук П.А.	М
2480	Кривич А.А.	М
2302	Павленко Е.А.	Ж
2500	Сокол Н.А.	Ж
3002	Павленко И.А.	М
2523	Павленко Т.Х.	Ж
2529	Хитрук А.П.	М
2570	Павленко П.И.	М



ID родителя	ID ребенка
2146	2302
2146	3002
2155	2302
2155	3002
2302	2431
2302	2511
2302	3193
3002	2586
3002	2570
2523	2586

Решение

3) В правой таблице ищем все пары **ID**, в которых первое число (в левой колонке) равно **2302** или **3002**. Это – внуки Кривич Л.П.

ID	Фамилия И.О.	Пол
2146	Кривич Л.П.	Ж
2155	Павленко А.К.	М
2431	Хитрук П.А.	М
2480	Кривич А.А.	М
2302	Павленко Е.А.	Ж
2500	Сокол Н.А.	Ж
3002	Павленко И.А.	М
2523	Павленко Т.Х.	Ж
2529	Хитрук А.П.	М
2570	Павленко П.И.	М

ID родителя	ID ребенка
2146	2302
2146	3002
2155	2302
2155	3002
2302	2431
2302	2511
2302	3193
3002	2586
3002	2570
2523	2586

Решение

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

ID	Фамилия И.О.	Пол
2146	Кривич Л.П.	Ж
2155	Павленко А.К.	М
2431	Хитрук П.А.	М
2480	Кривич А.А.	М
2302	Павленко Е.А.	Ж
2500	Сокол Н.А.	Ж
3002	Павленко И.А.	М
2523	Павленко Т.Х.	Ж
2529	Хитрук А.П.	М
2570	Павленко П.И.	М

ID родителя	ID ребенка
2146	2302
2146	3002
2155	2302
2155	3002
2302	2431
2302	2511
2302	3193
3002	2586
3002	2570
2523	2586

Ответ: 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

- 1) Разрешение равно 24 бит.
- 2) Частота дискретизации равна 64 кГц – 64000 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: $24 \times 64000 \times x \times 2 = 480 \cdot 2^{23}$.
Для удобства вычислений по возможности выделяем степени двойки.
Решаем уравнение, получаем: $x = 1310,72$ (секунд).
- 7) Переводим x в минуты: $x \approx 1311 / 60 \approx 21,85$ (минут).
- 8) Округляем до ближайшего значения, кратного 10: $x \approx 21,85 \approx 20$ (минут).

Ответ: 20.

По каналу связи передаются сообщения, содержащие только 4 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

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 или с началом ни 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.

На числовой прямой даны два отрезка: $P = [43, 49]$ и $Q = [44, 53]$.

Укажите наибольшую возможную длину отрезка A , 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

1) Как обычно, заменяем высказывания логическими переменными:

$$a = (x \in A), \quad p = (x \in P), \quad q = (x \in Q).$$

Тогда исходное логическое выражение принимает вид:

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

2) Избавимся от операции следования, используя закон 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 равны **1** в пределах соответствующих отрезков и **0** – вне их (причем границы входят в отрезки).

Решение

Анализируем полученное уравнение $\neg a \vee p \vee q = 1$:

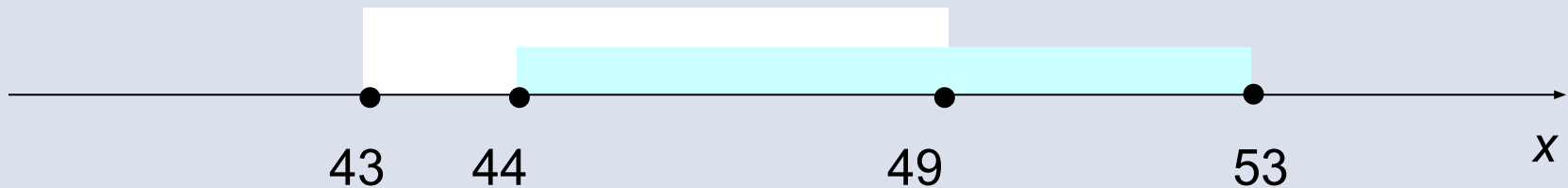
- переменные соединены операцией **ИЛИ**, следовательно, результат **1** будет автоматически обеспечен, если хотя бы одна из переменных – ***p*** или ***q*** – равна **1** (то есть везде, где числовая прямая «накрыта» хотя бы одним отрезком – ***P*** или ***Q***);
- там, где нет ни отрезка ***P***, ни 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***).

Решение

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

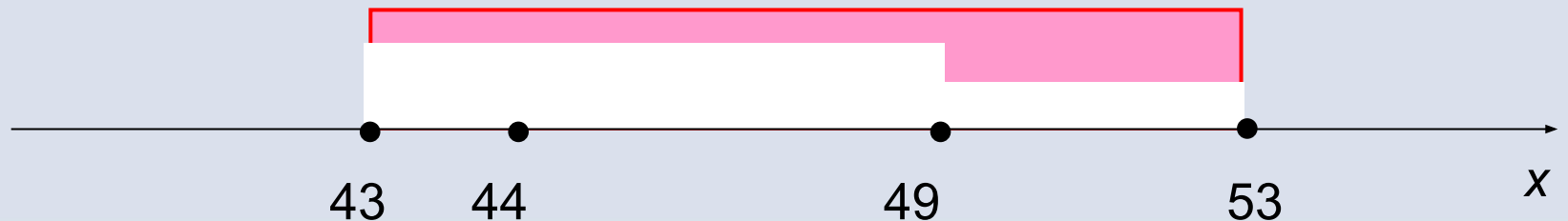
$$P = [43, 49]$$

$$Q = [44, 53]$$



5) Теперь 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**:

$$A = [43, 53]$$



Решение

б) А теперь нужно правильно вычислить длину полученного отрезка **А**, 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 **А** могут быть «выколоты» (могут не входить в отрезок), и тогда нужно при вычислении длины отрезка брать соседние натуральные граничные значения внутри отрезка. В нашем случае конечные точки **43** и **53** входят в отрезок **А**.

Теперь вспомним, как вычисляется длина отрезка на 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 **А** равна $53 - 43 = 10$.

Ответ: 10.

При регистрации в компьютерной системе каждому пользователю выдается пароль, состоящий из **8** 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

1) В алфавит входит **7** символов (**Z, X, C, V, B, N, M**).

При такой мощности алфавита для кодирования одного символа требуется **3 бита** ($2^3 = 8$, а $2^2 = 4$ – недостаточно).

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

3) Тогда для хранения одного пароля требуется **$24/8 = 3$ байта**.

(Если бы деление выполнялось не нацело, потребовалось бы взять ближайшее большее целое количество байт.)

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

Ответ: 75.

В программе описан одномерный целочисленный массив с индексами от **0** до **10**. Ниже представлен фрагмент программы, обрабатывающей данный массив:

```
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

1) Анализируем алгоритм:

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

Такое вычисление значения **s** можно расписать в одну строку в виде:

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

Раскроем скобки и перегруппируем слагаемые:

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

Как видим, большинство слагаемых сокращается, и получаем:

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

Решение

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: $A[0] = 999$, $A[10] = 101$ (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.

Система команд исполнителя РОБОТ, «живущего» в прямоугольном лабиринте на клетчатой плоскости, состоит из 8 команд. Четыре команды – это команды-приказы: **вверх**, **вниз**, **влево**, **вправо**. При выполнении любой из этих команд РОБОТ перемещается на одну клетку, соответственно, вверх, вниз, влево 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

1						
2						
3						
4						
5						
6						
	A	B	C	D	E	F

Решение

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: . Причем «правильной» является только одна такая клетка, а остальные две (позначим их красным цветом) – это «ловушки» для РОБОТа.

1						
2						
3						
4						
5						
6						
	A	B	C	D	E	F

Решение

Когда начинается очередной проход цикла, проверяется наличие стенки снизу. Если ее нет, то выполняется перемещение на 1 клетку вниз. Если же стенка снизу есть, то выполняется перемещение на 1 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.)

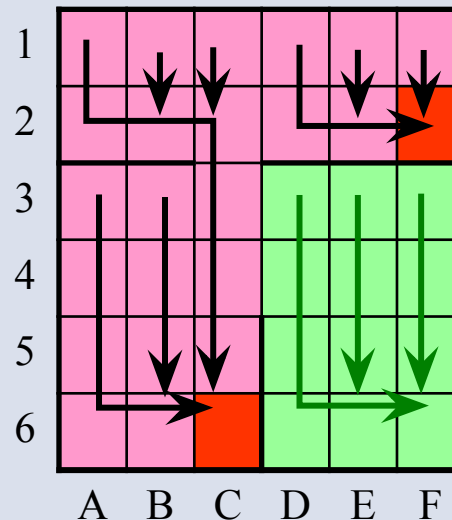
Решение

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».

Решение

Именно так мы и будем пытаться «трассировать» движение РОБОТа. Увидев, куда в итоге придет РОБОТ, помечаем все пройденные клетки розовым цветом (если попадаем в «ловушку»), или зеленым (если доходим до клетки **F6**). Сама клетка **F6** нам тоже подходит: если РОБОТ изначально находится в ней, то он просто никуда не пойдет.



Остается подсчитать количество клеточек, закрашенных зеленым: их **12**.

Ответ: 12.

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

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		

Решение

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		

Решение

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		

Решение

4) Тем самым мы получили «ключевое» уравнение, позволяющее вычислить x :

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

Решаем его как обычное квадратное уравнение:

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

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

$$x_{1,2} = (1 \pm 17) / (2 \cdot 3), \text{ тогда } x_1 = 18/6 = 3, x_2 = -16/6 = -22/3.$$

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

Ответ: 3.

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

Требуется проанализировать алгоритм работы программы и понять, какими могут быть значения введенного числа a .

- 1) Цикла 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, нужно 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) Нужно определить, какое возможное значение a является наименьшим. Так как неравенство нестрогое, это значение $a = 50$.

Ответ: 50.

Сколько существует различных символьных последовательностей длины 5 в трехбуквенном алфавите {К, О, Т}, которые содержат ровно две буквы О?

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:

ААААА

ААААО

АААОА

АААОО

ААОАА

...

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.

Решение

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:

О – 0, К – 1, Т – 2.

Почему мы выбрали нуль именно для буквы **О**?

Потому, что нам понадобится искать слова, содержащие ровно две буквы **О**, 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.

Решение

3) Первый нуль может быть в одной из **5** позиций – получаем **5** вариантов.

Как в каждом из этих вариантов может располагаться второй нуль?

- Когда первый нуль находится в позиции «1», второй нуль может располагаться в одной из 4 оставшихся позиций – «2», «3», «4», «5».
- Когда первый нуль находится в позиции «2», второй нуль может располагаться в одной из 3 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 2 оставшихся позиций правее – «4» и «5» (почему, – рассмотрено выше);
- когда первый нуль находится в позиции «4», второй нуль может располагаться только в 1 позиции правее – «5».

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

Решение

Возможны 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 разрядах – получаем 5 вариантов.

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 разноцветных нулей.

А теперь вспомним, что на самом деле оба нуля (синий и красный) – совершенно равноправны. 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**.

Решение

Итак, существует **10** вариантов размещения в числе двух нулей. В каждом из этих 10 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 10 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.

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

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 выполнялся $55/5 = 11$ раз.

Решение

3) Учитывая, что переменная s изначально тоже обнуляется, получаем, что за эти **11** проходов цикла переменная s увеличивается на $11 \cdot d$ и становится больше **365** (условие прекращения работы цикла).

Тогда получаем неравенство:

$$11d > 365,$$

откуда $d > 365/11$, т.е. $d > 332/11$, или, учитывая, что числа – целые, $d > 34$.

4) Чтобы теперь оценить верхнюю границу интервала возможных значений d , надо учесть, что при слишком большом значении d цикл мог бы прерваться еще после 10 проходов (а то и раньше). Значит, нам нужно определить значение d , на единицу меньше, чем минимальное значение d , обеспечивающее выполнение **10** проходов цикла.

В этом случае неравенство будет таким:

$$10d > 365,$$

тогда $d > 365/10$, или $d > 36,5$.

Решение

5) Рассуждая аналогично п. 3, можно заключить, что минимальное значение d , при котором цикл завершится через **10** проходов, равно **37** (ближайшее целое число, большее **36,5**). Нам же требуется значение d , на единицу меньше. 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.

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

$$F(n) = 1 \text{ при } n \leq 2;$$

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

Чему равно значение функции $F(8)$?

Решение

Это – задача на рекурсию. В условии этого задания отметим ошибку, делающую это условие внутренне противоречивым:

вторая строка, определяющая соотношения для функции $F(n)$, должна иметь вид: $F(n) = F(n - 2) \times (n + 2)$ при $n > 2$.

Решение

Само 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 $F(n)$:

$$F(1) = 1;$$

$$F(2) = 1;$$

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

$$F(4) = F(4 - 2) \times (4 + 2) = F(2) \times 6 = 1 \times 6 = 6;$$

$$F(5) = F(5 - 2) \times (5 + 2) = F(3) \times 7 = 5 \times 7 = 35;$$

$$F(6) = F(6 - 2) \times (6 + 2) = F(4) \times 8 = 6 \times 8 = 48;$$

$$F(7) = F(7 - 2) \times (7 + 2) = F(5) \times 9 = 35 \times 9 = 315;$$

$$F(8) = F(8 - 2) \times (8 + 2) = F(6) \times 10 = 48 \times 10 = 480.$$

Последнее вычисленное значение – и есть искомый ответ.

Ответ: 480.

Известно, что для целого числа x выполнено:

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

Определите значение x . Ответ запишите в десятичной системе счисления.

Решение

1) Запишем «каноническое» преобразование обоих недесятичных чисел в десятичную систему счисления (согласно формуле вида

$$a_1 a_{0n} = a_1 \cdot n^1 + a_0 \cdot n^0):$$

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

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

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

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

откуда $3x = 33$, тогда $x = 11$.

Ответ: 11.

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

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 накопления суммы значений $x \bmod 10$, а затем переменная x (исходное число) переписывается на значение $x \operatorname{div} 10$. Такой алгоритм – это разбор в цикле исходного числа x на отдельные цифры (справа налево), а в переменной b вычисляется сумма цифр исходного числа;
- после цикла выводится сначала значение переменной a , а затем – переменной b .

Решение

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

3) Для ответа на вопрос задачи нужно определить, сколько существует двузначных натуральных 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:

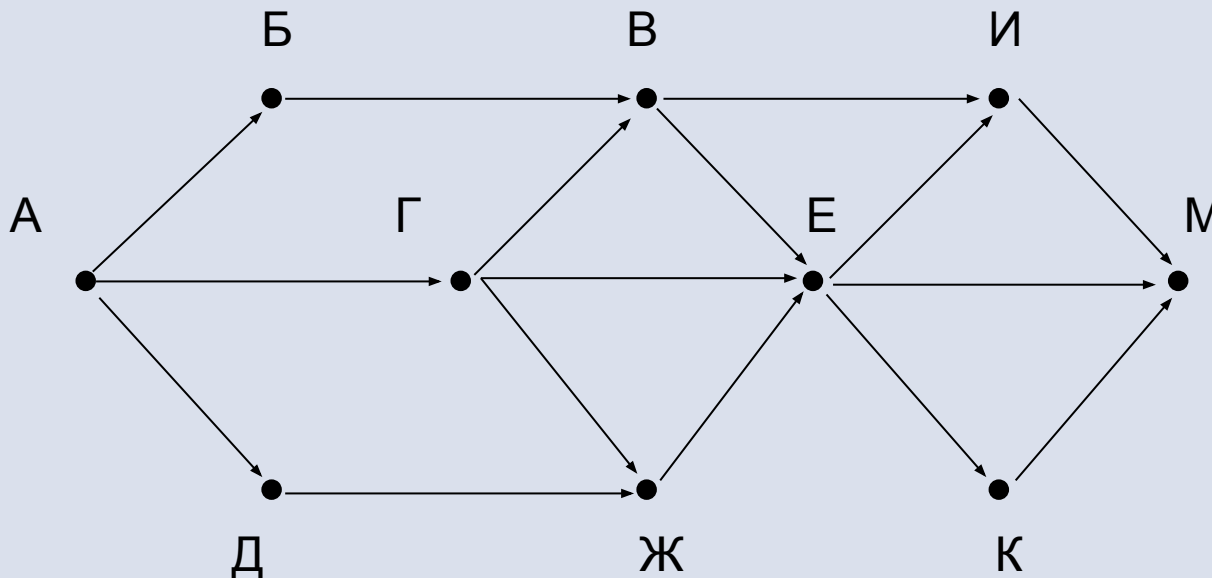
69, 96, 78, 87.

Всего – 4 числа.

Ответ: 4.

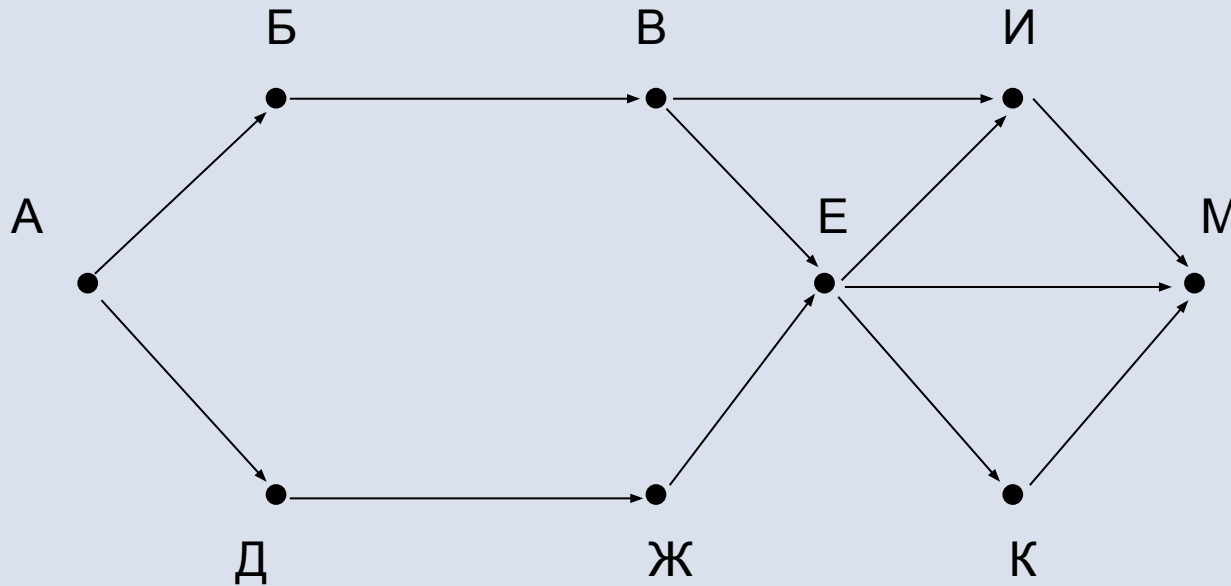
На рисунке – схема дорог, связывающих города **А**, **Б**, **В**, **Г**, **Д**, **Е**, **Ж**, **И**, **К**, **М**. По каждой дороге можно двигаться только в одном направлении, указанном стрелкой.

Сколько существует различных путей, ведущих из города **А** в город **М** и не проходящих через город **Г**?



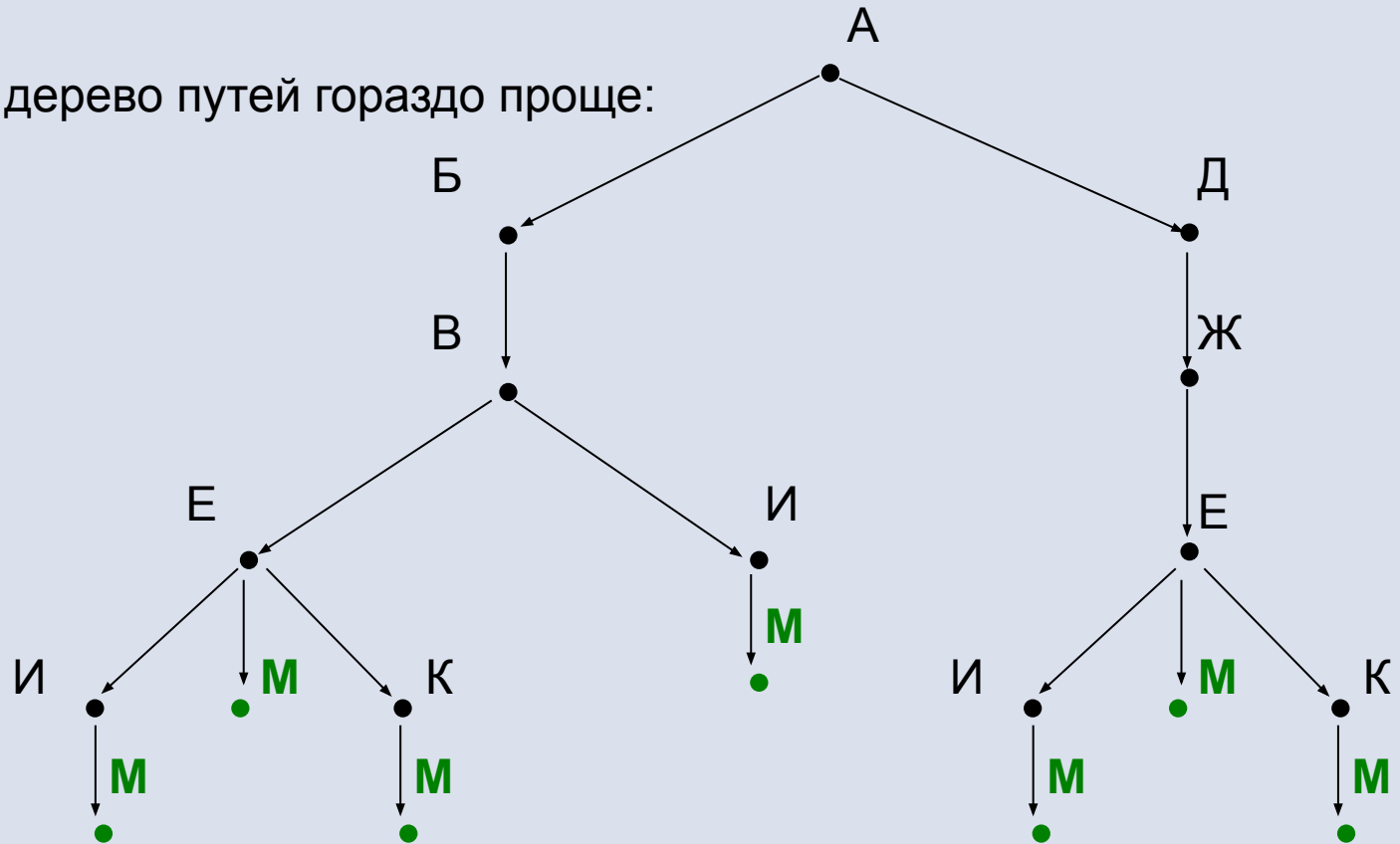
Решение

Существенно облегчим себе решение: раз через город Г идти нельзя, его можно сразу исключить из схемы.



Решение

Теперь строить дерево путей гораздо проще:



Количество конечных точек, соответствующих городу **М**, равно 7.

Ответ: 7.

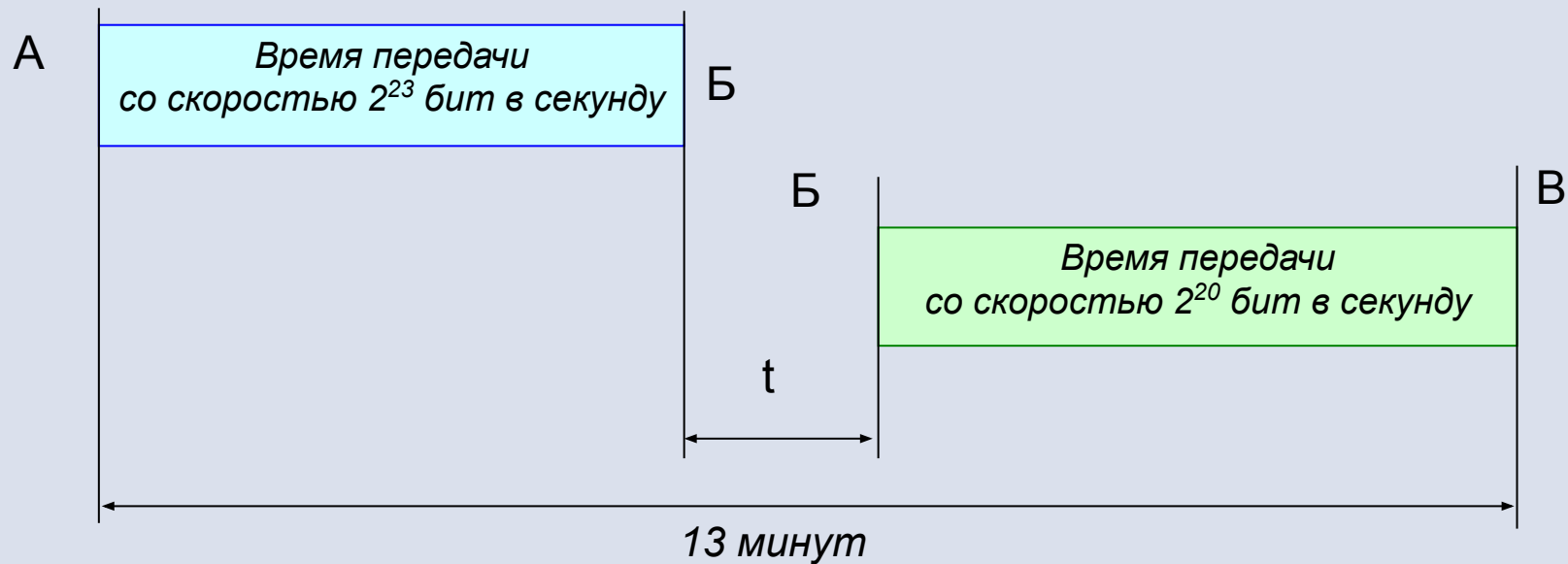
Данные объемом **80 Мбайт** передаются из пункта **А** в пункт **Б** по каналу связи, обеспечивающему скорость передачи данных **223 бит в секунду**, а затем из пункта **Б** в пункт **В** по каналу связи, обеспечивающему скорость передачи данных **220 бит в секунду**. От начала передачи данных из пункта **А** до их полного получения в пункте **В** прошло **13 минут**.

Через какое время в секундах началась передача данных в пункте **Б**, 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

1) Построим диаграмму Ганта для этой задачи:



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

Решение

2) Составляем по этой схеме уравнение (t – это время задержки между окончанием приема данных из **A** и началом их передачи в **B**).

Для этого вычисляем время передачи данных из **A** в **B** и из **B** в **B**:

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

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 **A** в **B** до начала их передачи из **B** в **B**, нужно к найденному значению t прибавить время передачи данных из **A** в **B**: $80 + 60 = 140 \text{ (с)}$.

Ответ: 140.

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

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.:

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

Биты маски нам неизвестны. 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**.

Решение

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:

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

Решение

- 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 искомый байт маски тоже равен **64**.

Ответ: 64.

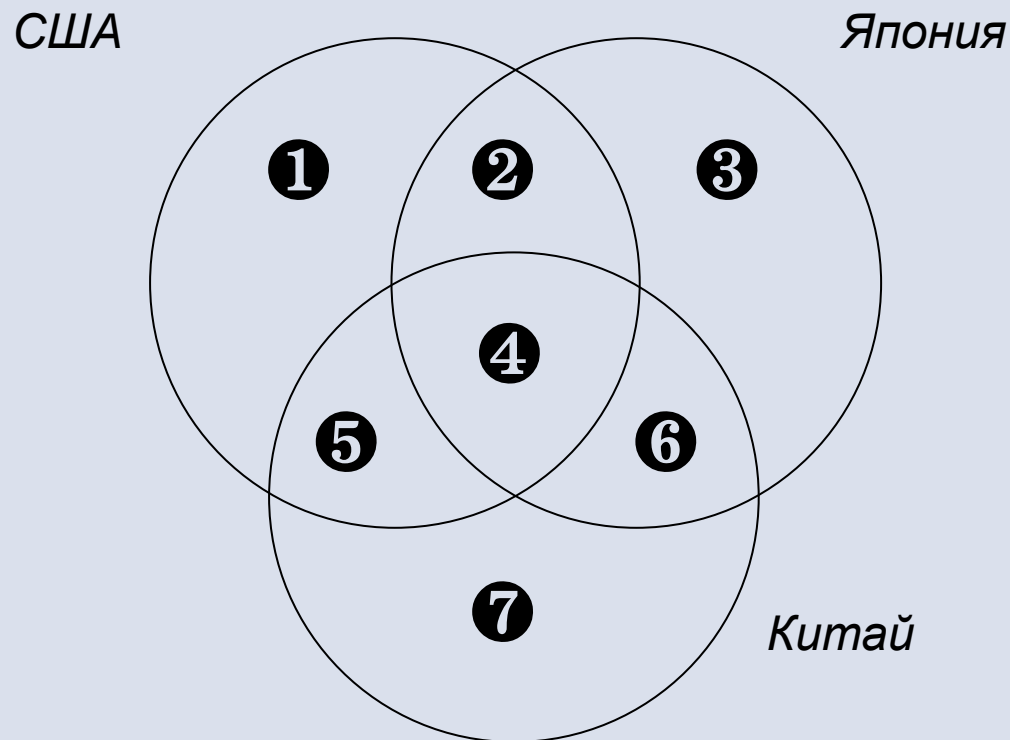
В языке запросов поискового сервера для обозначения логической операции «ИЛИ» используется символ «|», а для логической операции «И» – символ «&». В таблице приведены запросы и количество найденных по ним страниц некоторого сегмента сети Интернет.

Запрос	Найдено страниц (в тысячах)
<i>США Япония Китай</i>	450
<i>Япония Китай</i>	260
<i>(США & Япония) (США & Китай)</i>	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

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



Решение

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

США | Япония | Китай

Япония | Китай

(США & Япония) | (США & Китай) =
США & (Япония | Китай)

США

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

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

$$\mathbf{2} + \mathbf{4} + \mathbf{5} = 100$$

$$\mathbf{1} + \mathbf{2} + \mathbf{4} + \mathbf{5} = ?$$

Решение

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.

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

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

Прежде всего, заметим, что работа в языке программирования 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.

Решение

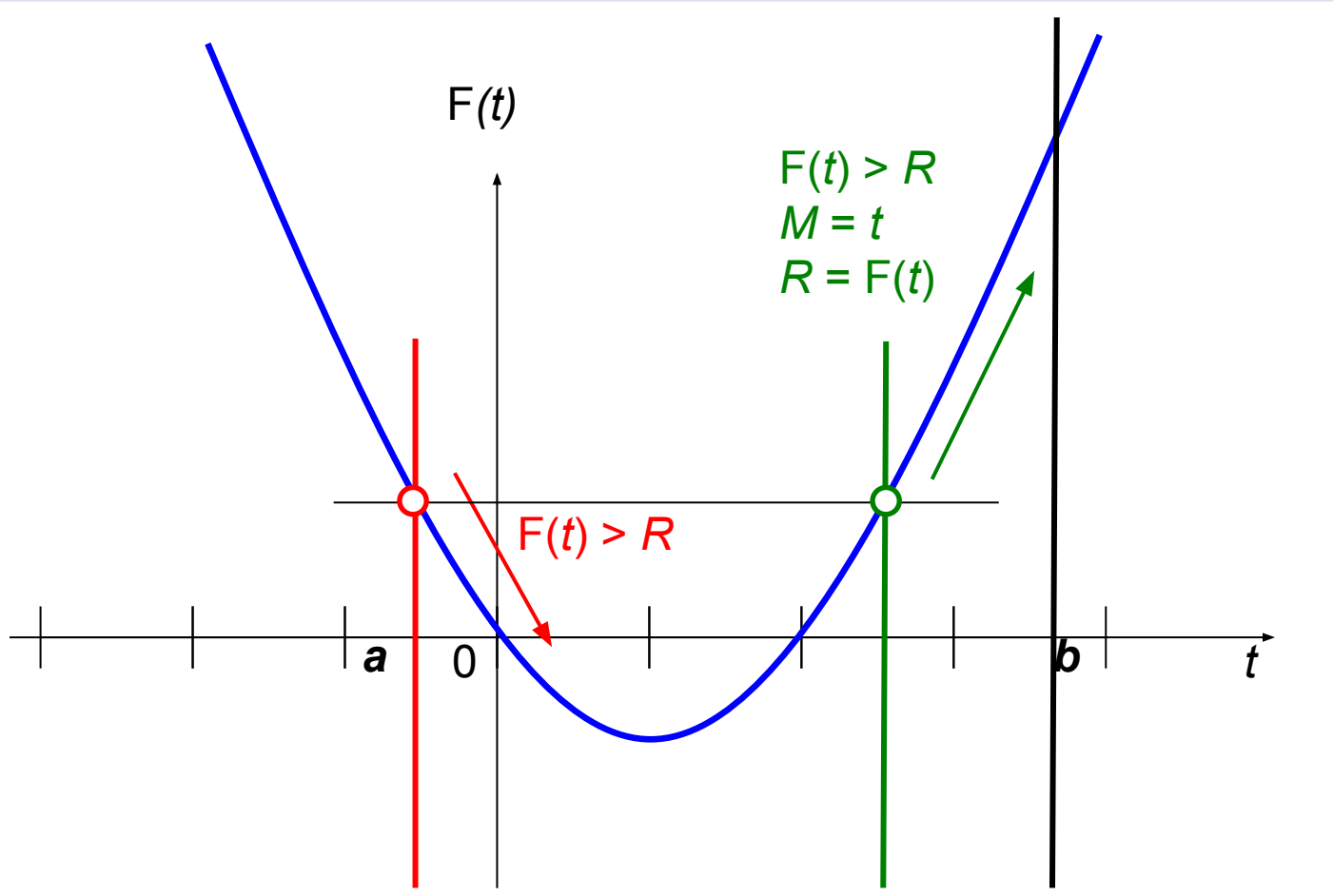
1) Анализируем алгоритм.

- 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), то текущее значение t заносится в переменную M , а значение функции в этой точке запоминается в R .
- До каких пор это будет продолжаться? Посмотрим на выражение, записанное в составе функции $F()$. Выполним в нем перемножение скобок. Получим:

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

Решение

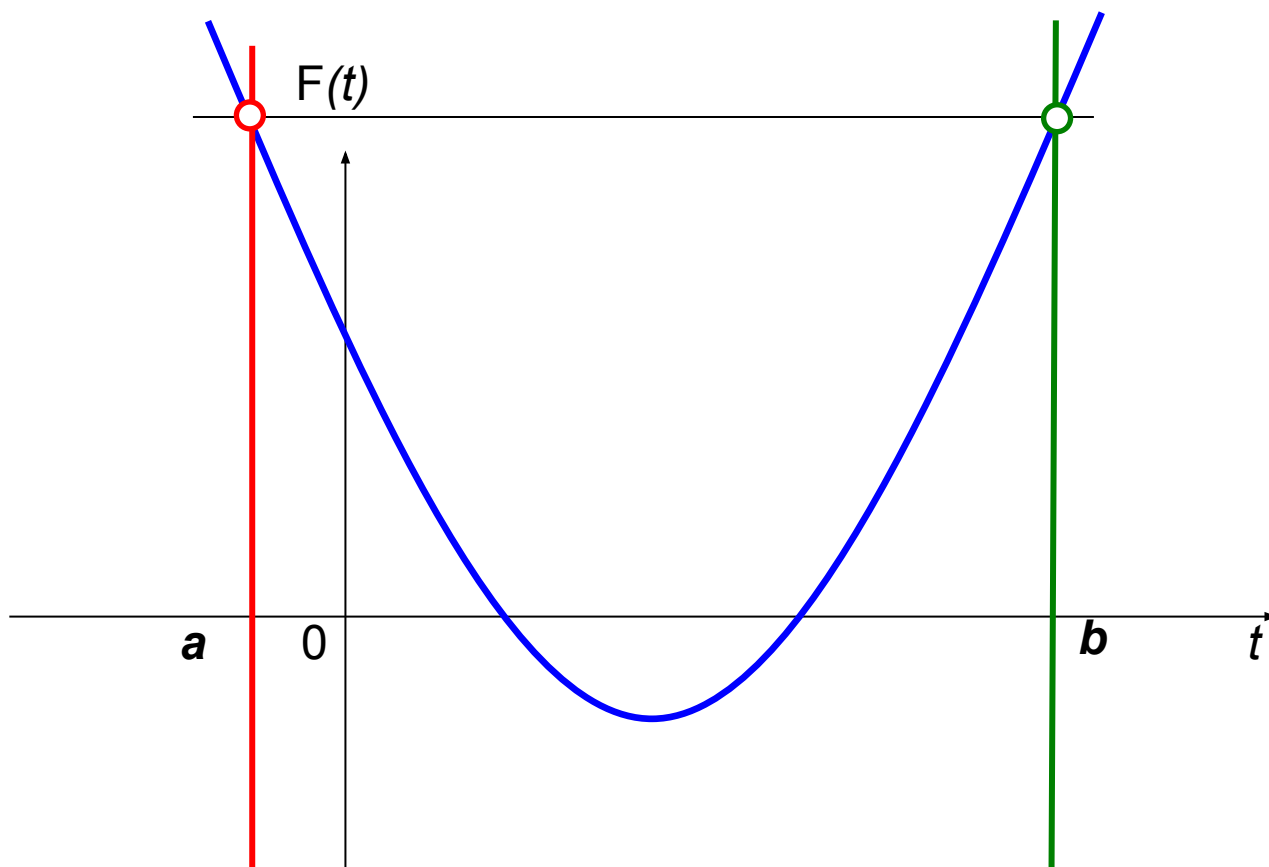
Это – зап
 графиком
 Поскольку
 И мы в се
 вниз. Зна
 не пройде
 и не подн
 $11t^2 - 22t$
 из основ
 в исходно
 параболу
 по дости
 значение
 либо, есл
 чем исход
 этому исх



лю),
 вверх.
 верху
 ока мы
 коде
 ачение
 oe
)
 io

Решение

2) Как ну
значен
Очевид
к увели
(которо
будет д
(и буде
Для эт
записа



ЖНОЕ

ЕНИЯ,

Решение

3) Остается вычислить значение H такое, чтобы получить нужное расположение параболы. Для этого надо составить уравнение в виде:

$$11a^2 - 22aH + (11H^2 + 13) = 11b^2 - 22bH + (11H^2 + 13)$$

или, подставляя известные нам значения $a = -10$ и $b = 50$:

$$11 \cdot (-10)^2 - 22 \cdot (-10)H + (11H^2 + 13) = 11 \cdot 50^2 - 22 \cdot 50H + (11H^2 + 13).$$

Очевидно, одинаковые слагаемые $(11H^2 + 13)$ сразу сокращаются. Также можно разделить обе его части на 11.

Получаем уравнение первого порядка:

$$(-10)^2 - 2 \cdot (-10)H = 50^2 - 2 \cdot 50H.$$

Решаем его:

$$\begin{aligned} 100 + 20H &= 2500 - 100H; & \Rightarrow & 100 + 20H = 2500 - 100H; \\ 120H &= 2400; & \Rightarrow & H = 20. \end{aligned}$$

Ответ: 20.