

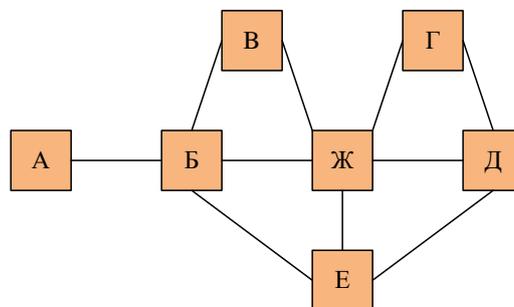
## Университетские субботы-2021. Информатика.

Юлия Борисовна Буркатовская, ОИТ

### Разбор демо-варианта 2021

**Задача 1. Анализ информационных моделей.** На рисунке схема дорог N-ского района изображена в виде графа, в таблице содержатся сведения о протяженности каждой из этих дорог (в километрах):

		Номер пункта						
		1	2	3	4	5	6	7
Номер пункта	1				9			7
	2				5		11	
	3						12	
	4	9	5			4	13	15
	5				4		10	8
	6		11	12	13	10		
	7	7			15	8		



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

**Решение.** Рассмотрим степени вершин (число дорог) в графе и в таблице.

Вершина	1	2	3	4	5	6	7
	2	2	1	5	3	4	3
Степень	А	Б	В	Г	Д	Е	Ж
	1	4	2	2	3	3	5

В графе есть три вершины с уникальными степенями: это Ж (степень 5), Б (степень 4) и А (степень 1). Согласно таблице, степень 5 имеет вершина 4, степень 4 – вершина 6, степень 1 – вершина 3. Итак, Ж=4, Б=6, А=3. Вершины В и Г имеют степень 2, значит, это вершины 1 и 2. Вершина В смежна с Б и Ж (то есть, с 6 и 4); согласно таблице, с 6 и 4 смежна вершина 2. Значит, В=2, Г=1. Вершина Г смежна с Ж и Д, вершина 1 смежна с 4 и 7, значит, Д=7, осталась вершина Е=5. Проверяем: Д смежна с Г, Ж, и Е в графе, 7 смежна с 1, 4 и 5 в таблице; Е смежна с Б, Ж, Д в графе, а 5 смежна с 6, 4 и 7 в таблице.

Итак, Г=1, Ж=4, расстояние между 1 и 4 равно 9.

**Ответ: 9.**

## Задание 2. Построение таблиц истинности логических выражений

Миша заполнял таблицу истинности функции  $(x \vee y) \wedge \neg(y \equiv z) \wedge \neg w$ , но успел заполнить лишь фрагмент из трех ее строк, даже не указав, какому столбцу таблицы соответствует какая из переменных  $w, x, y, z$ .

				$(x \vee y) \wedge \neg(y \equiv z) \wedge \neg w$
1		1		1
0	1		0	1
	1	1	0	1

Определите, какому столбцу таблицы истинности соответствует каждая из переменных  $w, x, y, z$ .

В ответе напишите буквы  $w, x, y, z$  в том порядке, в котором идут соответствующие им столбцы (сначала – буква, соответствующая первому столбцу; затем – буква, соответствующая второму столбцу, и т. д.). Буквы в ответе пишите подряд, никаких разделителей между буквами ставить не нужно.

*Пример.* Пусть задано выражение  $\neg x \vee y$ , зависящее от двух переменных  $x$  и  $y$ , и фрагмент таблицы истинности:

		$\neg x \vee y$
0	1	0

Тогда первому столбцу соответствует переменная  $y$ , а второму столбцу соответствует переменная  $x$ . В ответе нужно написать:  $yx$ .

**Решение.** Перепишем функцию в более удобном виде:

$$F = (x \vee y) \overline{(y \equiv z)} \bar{w}.$$

Заметим, что  $F = 1$  во всех строках таблицы. Конъюнкция равна 1, если и только если все ее аргументы равны 1, имеем систему:

$$\begin{aligned} x \vee y &= 1 \\ (y \equiv z) &= 0 \\ w &= 0 \end{aligned}$$

Согласно таблице,  $w$  может быть только в четвертом столбце, и весь столбец состоит из нулей.

			$w$	$(x \vee y) \overline{(y \equiv z)} \bar{w}$
1		1	0	1
0	1		0	1
	1	1	0	1

Из второго условия ясно, что  $y = \bar{z}$ , то есть,  $y$  и  $z$  занимают первый и второй столбцы (в первом и третьем, а также во втором и третьем есть совпадения). Значит,  $x$  находится в третьем столбце. Также в первом и втором столбце значения противоположны.

		$x$	$w$	$(x \vee y) \overline{(y \equiv z)} \bar{w}$
1	0	1	0	1
0	1		0	1
0	1	1	0	1

Строки должны быть различны, значит, во второй строке в столбце  $x$  будет 0. Из условия  $x \vee y = 1$  получаем, что  $y$  находится во втором столбце.

$z$	$y$	$x$	$w$	$(x \vee y) \overline{(y \equiv z)} \overline{w}$
1	0	1	0	1
0	1	0	0	1
0	1	1	0	1

Можно решать подобные задачи другим путем. Выпишем наборы значений переменных, на которых верна наша система уравнений.

$$\begin{aligned} x \vee y &= 1 \\ (y \equiv z) &= 0 \\ w &= 0 \end{aligned}$$

$x$	$y$	$z$	$w$	$(x \vee y) \overline{(y \equiv z)} \overline{w}$
1	0	1	0	1
0	1	0	0	1
1	1	0	0	1

Сравним с наборами в исходной таблице.

				$(x \vee y) \overline{(y \equiv z)} \overline{w}$
1		1		1
0	1		0	1
	1	1	0	1

Последний столбец – это  $w$  (все нули), первый –  $z$  (только там может быть два нуля). Тогда второй столбец –  $y$  (все значения инверсны  $z$ ), третий –  $x$ .

$z$	$y$	$x$	$w$	$(x \vee y) \overline{(y \equiv z)} \overline{w}$
1	0	1	0	1
0	1	0	0	1
0	1	1	0	1

**Ответ.**  $zuxw$ .

### Задание 3. Базы данных. Файловая система

Ниже представлены два фрагмента таблиц из базы данных о жителях микрорайона. Каждая строка таблицы 2 содержит информацию о ребенке и одном из его родителей. Информация представлена значением поля ID в соответствующей строке таблицы 1. Определите на основании приведенных данных ID женщины, ставшей матерью в наиболее молодом возрасте. При вычислении ответа учитывайте только информацию из приведенных фрагментов таблиц.

Таблица 1			
ID	Фамилия	Пол	Год рождения
14	Краснова И.А.	Ж	1937
24	Сканави И.П.	М	1943
25	Сканави П.И.	М	1974
26	Сканави П.П.	М	2001
34	Кушенко А.И.	Ж	1964
35	Кушенко В.С.	Ж	1990
36	Кушенко С.С.	М	1964
44	Лебедь А.С.	Ж	1938
45	Лебедь В.А.	М	1953
46	Гросс О.С.	Ж	1993
47	Гросс П.О.	М	2009
54	Клычко А.П.	Ж	1995
64	Крот П.А.	Ж	1973

Таблица 2	
ID родителя	ID ребенка
24	25
44	25
25	26
64	26
24	34
44	34
34	35
36	35
14	36
34	46
36	46
25	54
64	54

Решение. Нас интересуют матери, выделим в таблице 2 женщин в столбце родителей. В этих строках добавим год рождения родителей и детей. Вычислим возраст матери в момент рождения ребенка и выберем минимум. Это 22, ID матери 64.

Таблица 1			
ID	Фамилия	Пол	Год рождения
14	Краснова И.А.	Ж	1937
24	Сканави И.П.	М	1943
25	Сканави П.И.	М	1974
26	Сканави П.П.	М	2001
34	Кушенко А.И.	Ж	1964
35	Кушенко В.С.	Ж	1990
36	Кушенко С.С.	М	1964
44	Лебедь А.С.	Ж	1938
45	Лебедь В.А.	М	1953
46	Гросс О.С.	Ж	1993
47	Гросс П.О.	М	2009
54	Клычко А.П.	Ж	1995
64	Крот П.А.	Ж	1973

Таблица 2		
ID родителя	ID ребенка	Возраст матери
24	25	
44 (1938)	25 (1974)	36
25	26	
64 (1973)	26 (2001)	28
24	34	
44 (1938)	34 (1964)	26
34 (1964)	35 (1990)	26
36	35	
14 (1937)	36 (1964)	27
34 (1964)	46 (1993)	29
36	46	
25	54	
64 (1973)	54 (1995)	22

Ответ. 64.

#### Задание 4. Кодирование и декодирование информации

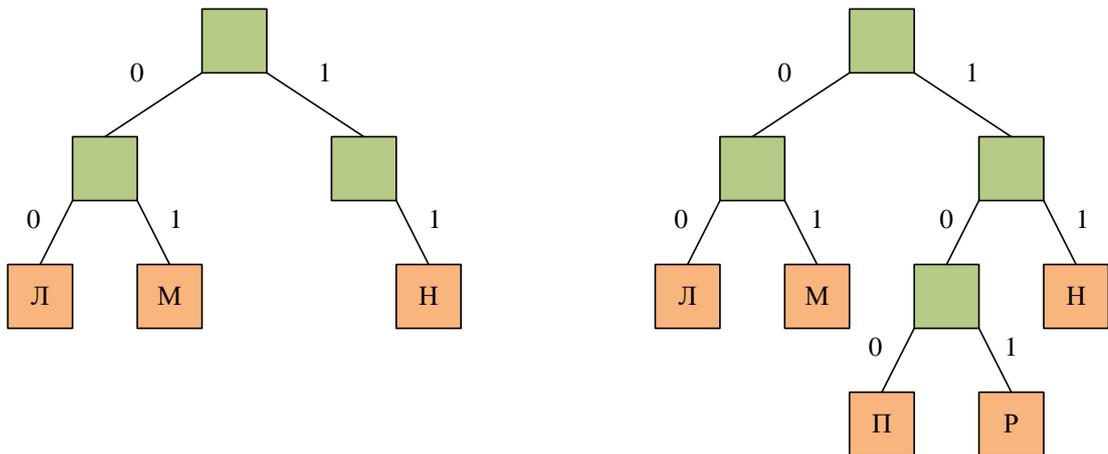
Для кодирования некоторой последовательности, состоящей из букв Л, М, Н, П, Р, решили использовать неравномерный двоичный код, удовлетворяющий условию, что никакое кодовое слово не является началом другого кодового слова. Это условие обеспечивает возможность однозначной расшифровки закодированных сообщений. Для букв Л, М, Н использовали соответственно кодовые слова 00, 01, 11. Для двух оставшихся букв – П и Р – кодовые слова неизвестны.

Укажите кратчайшее кодовое слово для буквы П, при котором код будет удовлетворять условию. Если таких слов несколько, укажите код с наименьшим числовым значением.

**Решение.** Удобно коды с таким условием (условие Фано) представлять в виде бинарного дерева, ребра помечать символами 0 (левое) и 1 (правое), а буквы располагать только в листьях. Код составляется из последовательности меток ребер от корня до листа.

Слева показано дерево заданных кодов. Новые коды можно получить, только проведя ребро налево от кода 1, тогда коды будут начинаться с 10. Учтем, что нужно закодировать еще две буквы, а буквы находятся в листьях. То есть после 10 делаем еще ветвление, получаем коды 100 и 101. Для П нужен код с меньшим числовым значением, это 100. Итоговое дерево показано справа.

Отметим, что взять код 10 для П нельзя, нечем будет закодировать Р.



**Ответ. 100.**

### Задание 5. Анализ и построение алгоритмов для исполнителей

На вход алгоритма подаётся натуральное число  $N$ . Алгоритм строит по нему новое число  $R$  следующим образом.

- 1) Строится двоичная запись числа  $N$ .
- 2) К этой записи дописываются справа ещё два разряда по следующему правилу:
  - а) складываются все цифры двоичной записи, и остаток от деления суммы на 2 дописывается в конец числа (справа). Например, запись 11100 преобразуется в запись 111001;
  - б) над этой записью производятся те же действия — справа дописывается остаток от деления суммы цифр на 2.

Полученная таким образом запись (в ней на два разряда больше, чем в записи исходного числа  $N$ ) является двоичной записью искомого числа  $R$ .

Укажите такое наименьшее число  $N$ , для которого результат работы алгоритма превышает 77. В ответе это число запишите в десятичной системе.

**Решение.** Заметим, что после шага 2а число единиц в двоичной записи числа будет четным, поэтому на шаге 2б в конец числа допишется 0. В итоге мы получим четное число (оканчивается в двоичном виде на 0), в двоичной записи которого четное число единиц. Рассмотрим четные числа, большие 77, и найдем наименьшее такое число.

$$78(10) = 64 + 8 + 4 + 2 = 1001110(2)$$

Первое же число подошло,  $R = 1001110(2)$ . Чтобы получить из него число  $N$ , отбросим две последних цифры.

$$N = 10011(2) = 16 + 2 + 1 = 19(10).$$

Также можно использовать факт, что отбрасывание двух последних цифр двоичной записи равносильно целочисленному делению десятичного числа на 4, то есть

$$N = 78/4 = 19.$$

**Ответ. 19.**

## Задача 6. Анализ программ

Определите, при каком наибольшем введенном значении переменной  $s$  программа выведет число 64. Для вашего удобства программа представлена на четырех языках программирования.

Pascal	Python
<pre>vars, n: integer; begin readln(s);   s:=s div 10;   n:=1;   while s&lt;51 do   begin     s:=s+5;     n:=n*2;   end; writeln(n); end.</pre>	<pre>s=int(input()) s=s//10 n=1 while s&lt;51:     s=s+5     n=n*2 print(n)</pre>
C++	Алгоритмический язык
<pre>#include &lt;iostream&gt; using namespace std; int main() { int s, n; cin&gt;&gt;s;   s=s/10;   n=1;   while (s&lt;51)   {     s=s+5;     n=n*2;   } cout&lt;&lt;n&lt;&lt;endl; return 0; }</pre>	<pre>алг нач цел n, s ввод s   s:=div(s,10)   n:=1 нц пока s &lt; 51   s:=s+5 n:=n*2 кц вывод n кон</pre>

**Решение.** В начале цикла  $n = 1$ , каждый раз в цикле  $n$  умножается на 2, это означает, что, если цикл выполнялся  $k$  раз,  $n = 2^k$ . К  $s$  каждый раз прибавляется 5, значит, в конце цикла ее значение будет  $s + 5k$ . Выход из цикла осуществляется по переменной  $s$ , условие  $s + 5k < 51$  ложно, но на предыдущем шаге программа не вышла из цикла, значит,  $s + 5(k - 1) < 51$  истинно. Итак,

$$n = 2^k = 64 \Rightarrow k = 6$$

$$s + 5k = s + 30 \geq 51 \Rightarrow s \geq 21$$

$$s + 5(k - 1) = s + 25 < 51 \Rightarrow s < 26$$

Получено условие на  $s$ :

$$21 \leq s \leq 25.$$

Вспомним, что введенное значение  $s$  было поделено на 10, значит, до деления

$$210 \leq s \leq 259.$$

Здесь учтено, что деление целочисленное. Наибольшее значение 259.

**Ответ. 259.**

### Задание 7. Кодирование и декодирование информации. Передача информации

Для хранения произвольного растрового изображения размером  $128 \times 320$  пикселей отведено 20 Кбайт памяти без учета размера заголовка файла. Для кодирования цвета каждого пикселя используется одинаковое количество бит, коды пикселей записываются в файл один за другим без промежутков. Какое максимальное количество цветов можно использовать в изображении?

**Решение.** Пусть  $k$  бит используется для каждого пикселя, тогда общий объем памяти в битах равен  $128 \times 320 \times k$ . Он не должен превышать 20 Кбайт, то есть  $20 \times 2^{13}$  бит.

$$128 \times 320 \times k \leq 20 \times 2^{13}$$

$$128 \times 16 \times k \leq 2^{13}$$

$$k \leq 2^2 = 4.$$

Используя 4 бита, можно закодировать  $2^4 = 16$  цветов.

**Ответ. 16.**

### **Задание 8. Перебор слов и системы счисления**

Игорь составляет таблицу кодовых слов для передачи сообщений, каждому сообщению соответствует своё кодовое слово. В качестве кодовых слов Игорь использует трехбуквенные слова, в которых есть только буквы Ш, К, О, Л, А, причём буква К появляется ровно 1 раз. Каждая из других допустимых букв может встречаться в кодовом слове любое количество раз или не встречаться совсем. Сколько различных кодовых слов может использовать Игорь?

**Решение.** Буква К может появиться на любой позиции в слове, это три варианта. В каждом из этих вариантов на остальных двух позициях может появиться любая из четырех оставшихся букв, это  $4^2 = 16$  вариантов. Итого получаем  $3 \times 16 = 48$  слов.

**Ответ. 48.**

### **Задание 9. Эксель**

Откройте файл электронной таблицы, содержащей вещественные числа — результаты ежечасного измерения температуры воздуха на протяжении трёх месяцев. Найдите разность между максимальным значением температуры и её средним арифметическим значением. В ответе запишите только целую часть получившегося числа.

**Решение.** Используем статистические функции МАКС(B2:Y92) и СРЗНАЧ(B2:Y92), вычисляем их разность. Целая часть от разности равна 14. Решение приведено в файле.

**Ответ. 14.**

### **Задание 10. Поиск слов в текстовом редакторе.**

С помощью текстового редактора определите, сколько раз, не считая сносок, встречается слово «долг» или «Долг» в тексте романа в стихах А. С. Пушкина «Евгений Онегин». Другие формы слова «долг», такие как «долги», «долгами» и т. д., учитывать не следует. В ответе укажите только число.

**Решение.** Можно воспользоваться вкладкой «Найти», для упрощения открыть вкладку «Больше» и отметить «Слово целиком», «Учитывать регистр» не отмечать. Потом пользоваться кнопкой «Найти далее».

Также можно пользоваться вкладкой «Заменить все» с теми же параметрами. После выполнения замены выведется сообщение, сколько замен произведено.

**Ответ. 1.**

### Задание 11. Вычисление количества информации

При регистрации в компьютерной системе каждому объекту сопоставляется идентификатор, состоящий из 15 символов и содержащий только символы из 8-символьного набора: А, В, С, D, E, F, G, H. В базе данных для хранения сведений о каждом объекте отведено одинаковое и минимально возможное целое число байт. При этом используют посимвольное кодирование идентификаторов, все символы кодируют одинаковым и минимально возможным количеством бит. Кроме собственно идентификатора, для каждого объекта в системе хранятся дополнительные сведения, для чего отведено 24 байта на один объект.

Определите объём памяти (в байтах), необходимый для хранения сведений о 20 объектах. В ответе запишите только целое число — количество байт.

#### Решение.

- Для кодирования 8 символов из заданного требуется минимум 3 бита ( $2^3 = 8$ ).
- В идентификаторе 15 символов, всего требуется  $15 \times 3 = 45$  бит.
- Под идентификатор отводят минимальное целое число байт  $k$ , находим его из условия  $45 \leq 8k$ , минимальное такое значение  $k = 6$ .
- Добавляем дополнительные сведения, итого  $6 + 24 = 30$  байт.
- Для 20 объектов нужно  $20 \times 30 = 600$  байт.

**Ответ. 600.**

## Задание 12. Выполнение алгоритмов для исполнителей

Исполнитель Редактор получает на вход строку цифр и преобразовывает её. Редактор может выполнять две команды, в обеих командах  $v$  и  $w$  обозначают цепочки цифр.

А) **заменить** ( $v, w$ ).

Эта команда заменяет в строке первое слева вхождение цепочки  $v$  на цепочку  $w$ . Например, выполнение команды

**заменить** (111, 27)

преобразует строку 05111150 в строку 0527150. Если в строке нет вхождений цепочки  $v$ , то выполнение команды **заменить** ( $v, w$ ) не меняет эту строку.

Б) **нашлось** ( $v$ ).

Эта команда проверяет, встречается ли цепочка  $v$  в строке исполнителя Редактор. Если она встречается, то команда возвращает логическое значение «истина», в противном случае возвращает значение «ложь». Строка исполнителя при этом не изменяется.

Цикл

```
ПОКА условие
  последовательность команд
КОНЕЦ ПОКА
```

выполняется, пока условие истинно.

В конструкции

```
ЕСЛИ условие
  ТО команда1
  ИНАЧЕ команда2
КОНЕЦ ЕСЛИ
```

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

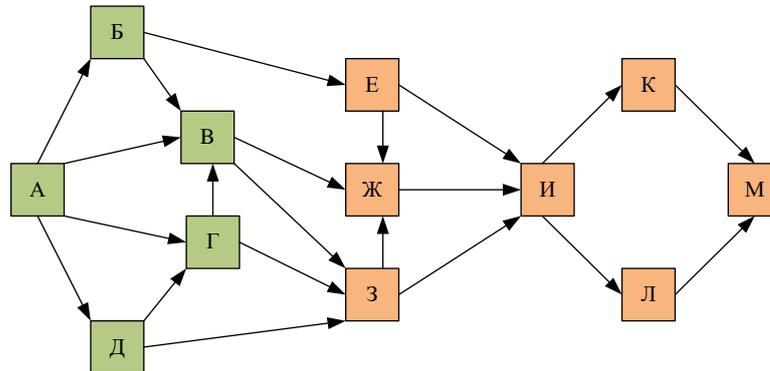
Какая строка получится в результате применения приведённой ниже программы к строке, состоящей из 70 идущих подряд цифр 8? В ответе запишите полученную строку.

```
НАЧАЛО
ПОКА нашлось (2222) ИЛИ нашлось (8888)
  ЕСЛИ нашлось (2222)
    ТО заменить (2222, 88)
    ИНАЧЕ заменить (8888, 22)
  КОНЕЦ ЕСЛИ
КОНЕЦ ПОКА
КОНЕЦ
```



### Задание 13. Поиск путей в графе

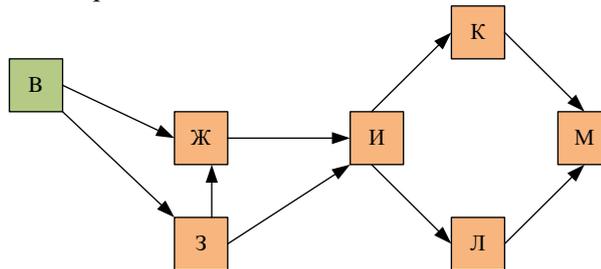
На рисунке представлена схема дорог, связывающих города А, Б, В, Г, Д, Е, Ж, З, И, К, Л, М. По каждой дороге можно двигаться только в одном направлении, указанном стрелкой. Сколько существует различных путей из города А в город М, проходящих через город В?



**Решение.** Проведем поиск путей в два этапа, сначала до обязательной вершины В. Составим уравнения. Решим их «снизу вверх», учитывая, что  $A=1$ .

$$\begin{aligned}V &= A + B + G = 4 \\ B &= A = 1 \\ G &= A + D = 2 \\ D &= A = 1\end{aligned}$$

Найдем теперь пути от В до М. Проанализируем граф, идем из вершины В, рассматриваем только вершины, куда можно попасть из В. Это Ж и З, из них попадаем в И, далее в К и Л, потом в М. Оставляем в графе только эти вершины.



Составляем систему уравнений, которую сразу можно решать «сверху вниз».

$$\begin{aligned}Z &= V = 4 \\ Zh &= Z + V = 8 \\ I &= Zh + Z = 12 \\ K &= I = 12 \\ L &= I = 12 \\ M &= K + L = 24.\end{aligned}$$

**Ответ. 24.**

**Задание 14. Кодирование чисел. Системы счисления**

Значение арифметического выражения:  $49^7 + 7^{21} - 7$  записали в системе счисления с основанием 7. Сколько цифр 6 содержится в этой записи?

**Решение.** Приведем все к степеням 7.

$$49^7 + 7^{21} - 7 = 7^{14} + 7^{21} - 7$$

При записи в системе с основанием 7 число  $7^k$  есть  $10\dots 0$  ( $k$  нулей). Рассмотрим  $7^{14} - 7$ , получим

$$\begin{array}{r} \_ 1000000000000000 \\ \phantom{\_ 1000000000000000} 10 \\ \hline 666666666666660 \end{array}$$

В этой записи 13 цифр 6. Прибавляем  $7^{21}$ , получаем число, в котором по-прежнему 13 цифр 6.

$$100000006666666666660$$

**Ответ.13.**

### Задание 15. Преобразование логических выражений

Обозначим через ДЕЛ( $n$ ,  $m$ ) утверждение «натуральное число  $n$  делится без остатка на натуральное число  $m$ ».

Для какого наибольшего натурального числа  $A$  формула

$$\neg \text{ДЕЛ}(x, A) \rightarrow (\text{ДЕЛ}(x, 6) \rightarrow \neg \text{ДЕЛ}(x, 9))$$

тождественно истинна (то есть принимает значение 1 при любом натуральном значении переменной  $x$ )?

#### Решение.

- Более удобные обозначения

$$\overline{(x|A)} \rightarrow ((x|6) \rightarrow \overline{(x|9)}) = 1$$

- Перейдем к дизъюнктивной форме

$$(x|A) \vee \overline{(x|6)} \vee \overline{(x|9)} = 1$$

- Нас интересует случай

$$\begin{cases} \overline{(x|6)} \vee \overline{(x|9)} = 0 \\ (x|A) = 1 \end{cases}$$

- Система

$$\begin{cases} (x|6) = 1 \\ (x|9) = 1 \\ (x|A) = 1 \end{cases}$$

То есть все числа, кратные 6 и 9, должны быть кратны  $A$ .  $A$  – максимально (иначе, не решая уравнение вообще, можно взять  $A=1$ ). Наименьшее число, кратное 6 и 9, это 18, оно должно быть кратно  $A$ , т.е.,  $A$  – делитель 18. Наибольший такой делитель это 18.

**Ответ. 18.**

### Задание 16. Рекурсивные алгоритмы

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

$$F(n) = 1 \text{ при } n = 1;$$

$$F(n) = n + F(n - 1), \text{ если } n \text{ – чётно,}$$

$$F(n) = 2 \times F(n - 2), \text{ если } n > 1 \text{ и при этом } n \text{ – нечётно.}$$

Чему равно значение функции  $F(26)$ ?

**Решение.** Записываем формулы по описанию функции, «сверху вниз»:

$$\begin{aligned} F(26) &= 26 + F(25) \\ F(25) &= 2F(23) = 4F(21) = \dots \end{aligned}$$

Далее аргументы у функции нечетны, потому что каждый раз от аргумента вычитается 2. Известно  $F(1)$ , то есть, рекурсивно будем вычислять функцию, пока не получим аргумент 1. Начальный аргумент – 25, каждый раз при рекурсии вычитается 2, вычтем 12 раз. Каждый раз умножаем результат на 2, это тоже произойдет 12 раз:

$$\begin{aligned} F(25) &= 2F(23) = 4F(21) = \dots = 2^{12}F(1) = 2^{12} = 4096 \\ F(26) &= 26 + F(25) = 26 + 4096 = 4122 \end{aligned}$$

**Ответ. 4122.**

### Задание 17. Проверка на делимость

Рассматривается множество целых чисел, принадлежащих числовому отрезку [1016; 7937], которые делятся на 3 и не делятся на 7, 17, 19, 27. Найдите количество таких чисел и максимальное из них. В ответе запишите два целых числа: сначала количество, затем максимальное число.

Для выполнения этого задания можно написать программу или воспользоваться редактором электронных таблиц.

**Решение.** Здесь учтен признак делимости на 3, можно взять первое число из промежутка, кратное 3, а затем прибавлять к нему 3. Тогда делимость на 3 можно не проверять.

Pascal
<pre>var k,m,n: integer; begin   k:=0;   n:=1017;   while(n&lt;=7937) do   begin     if((n mod 7&gt;0)and(n mod 17&gt;0)and(n mod 19&gt;0)and(n mod 27&gt;0)) then     begin       k:=k+1;       m:=n;     end;     n:=n+3;   end;   writeln(k, ' ',m); end.</pre>
C++
<pre>#include &lt;iostream&gt; using namespace std; int main() { int n,k=0,m;   for(n=1017;n&lt;=7937;n+=3)     if((n%7&gt;0) &amp;&amp; (n%17&gt;0) &amp;&amp; (n%19&gt;0) &amp;&amp; (n%27&gt;0))     {       k++;       m=n;     }   cout&lt;&lt;k&lt;&lt;' '&lt;&lt;m; }</pre>
Python
<pre>k=0 for n in range(1017,7938,3):   if n%7&gt;0 and n%17&gt;0 and n%19&gt;0 and n%27&gt;0:     k+=1     m=n print(k,m)</pre>

**Ответ. 1568 7935.**

### Задание 18. Робот-сборщик монет.

Квадрат разлинован на  $N \times N$  клеток ( $1 < N < 17$ ). Исполнитель Робот может перемещаться по клеткам, выполняя за одно перемещение одну из двух команд: вправо или вниз. По команде вправо Робот перемещается в соседнюю правую клетку, по команде вниз — в соседнюю нижнюю. При попытке выхода за границу квадрата Робот разрушается. Перед каждым запуском Робота в каждой клетке квадрата лежит монета достоинством от 1 до 100. Посетив клетку, Робот забирает монету с собой; это также относится к начальной и конечной клетке маршрута Робота.

Откройте файл. Определите максимальную и минимальную денежную сумму, которую может собрать Робот, пройдя из левой верхней клетки в правую нижнюю. В ответ запишите два числа друг за другом без разделительных знаков — сначала максимальную сумму, затем минимальную.

Исходные данные представляют собой электронную таблицу размером  $N \times N$ , каждая ячейка которой соответствует клетке квадрата.

*Пример входных данных:*

1	8	8	4
10	1	1	3
1	3	12	2
2	3	5	6

Для указанных входных данных ответом должна быть пара чисел 41 и 22.

**Решение.** Решение находится в приложенном файле.

- В ячейку A12 заносим формулу =A1
- В ячейку B12 заносим формулу =A12+B1. Копируем формулу в диапазон C12:J12.
- В ячейку A13 заносим формулу =A12+A2. Копируем формулу в диапазон A14:A21.
- В ячейку B13 заносим формулу =МАКС(B12;A13)+B2. Копируем формулу в диапазон B13:J21.
- Первое число ответа – число в ячейке J21.
  
- В ячейку A23 заносим формулу =A1
- В ячейку B23 заносим формулу =A23+B1. Копируем формулу в диапазон C23:J23.
- В ячейку A24 заносим формулу =A23+A2. Копируем формулу в диапазон A25:A32.
- В ячейку B24 заносим формулу =МАКС(B23;A24)+B2. Копируем формулу в диапазон B24:J32.
- Второе число ответа – число в ячейке J32.

**Ответ. 1204 502.**

### Задание 19. Выигрышная стратегия. Задание 1

Два игрока, Петя и Ваня, играют в следующую игру. Перед игроками лежат две кучи камней. Игроки ходят по очереди, первый ход делает Петя. За один ход игрок может добавить в одну из куч (по своему выбору) один камень или увеличить количество камней в куче в два раза. Например, пусть в одной куче 10 камней, а в другой 5 камней; такую позицию в игре будем обозначать  $(10,5)$ . Тогда за один ход можно получить любую из четырёх позиций:  $(11,5)$ ,  $(20,5)$ ,  $(10,6)$ ,  $(10,10)$ . Для того чтобы делать ходы, у каждого игрока есть неограниченное количество камней.

Игра завершается в тот момент, когда суммарное количество камней в кучах становится не менее 77. Победителем считается игрок, сделавший последний ход, т.е. первым получивший такую позицию, при которой в кучах будет 77 или больше камней.

В начальный момент в первой куче было семь камней, во второй куче –  $S$  камней;  $1 \leq S \leq 69$ .

Будем говорить, что игрок имеет выигрышную стратегию, если он может выиграть при любых ходах противника. Описать стратегию игрока – значит описать, какой ход он должен сделать в любой ситуации, которая ему может встретиться при различной игре противника. В описание выигрышной стратегии не следует включать ходы играющего по этой стратегии игрока, не являющиеся для него безусловно выигрышными, т.е. не являющиеся выигрышными независимо от игры противника.

Известно, что Ваня выиграл своим первым ходом после неудачного первого хода Пети. Укажите минимальное значение  $S$ , когда такая ситуация возможна.

**Решение.** Если Петя сделал неудачный ход, это значит, что он добавил в кучу слишком много камней, что позволило выиграть Ване. Ясно, что в этом случае,  $S > 7$ , иначе после Петиного хода в кучах будет максимум  $7 + 2 \times 7 = 21$  камень (позиция  $(7,14)$ ), и Ваня первым ходом не выигрывает. Мы ищем минимальное значение  $S$ , то есть Петя неудачным ходом максимально увеличил число камней и получил позицию  $(7,2S)$ , а Ваня – позицию  $(7,4S)$ , причем

$$7 + 4S \geq 77 \Rightarrow S \geq 18.$$

**Ответ. 18.**

,

## Задание 20. Выигрышная стратегия. Задание 2

Для игры, описанной в предыдущем задании, найдите два таких значения  $S$ , при которых у Пети есть выигрышная стратегия, причём одновременно выполняются два условия:

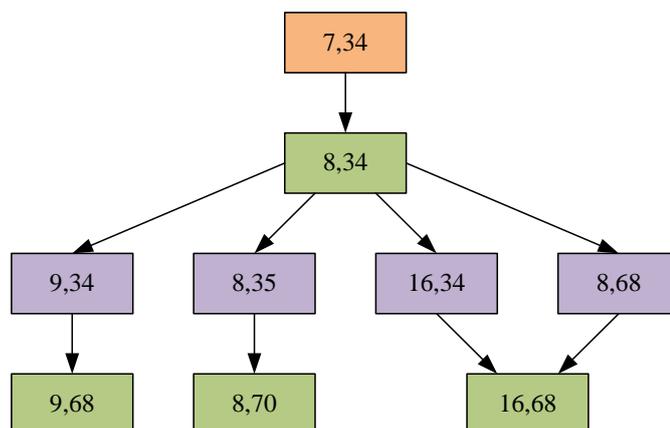
- Петя не может выиграть за один ход;
- Петя может выиграть своим вторым ходом независимо от того, как будет ходить Ваня.

Найденные значения запишите в ответе в порядке возрастания.

**Решение.** Найдем сначала позиции, из которых игрок может выиграть за один ход. Мы уже определились, что  $S > 7$ , то есть мы максимально увеличим число камней, если удвоим их количество во второй куче. При этом мы выигрываем при условии

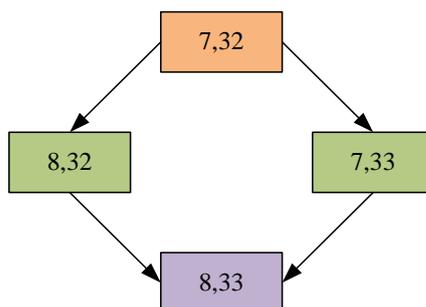
$$7 + 2S \geq 77 \Rightarrow S \geq 35$$

Получается, что при  $S \leq 34$  выиграть первым ходом невозможно, и нас интересуют именно такие значения. Возьмем  $S = 34$ . Чтобы выиграть, Пете нужно минимально увеличить число камней, то есть добавить один камень (причем в первую кучу, иначе выиграет Ваня). Рассмотрим этот ход Пети и все ходы Вани, для каждого хода Вани проверим, существует ли у Пети выигрышный ход. Ходы Пети выделены зеленым, ходы Вани – сиреневым.



Как видно, у Пети есть выигрышный ход для любой из позиций, иногда не единственный. То есть,  $S = 34$  удовлетворяет условиям.

Найдем второе число. Попутно мы выяснили, что позиция  $(8,34)$  является проигрышной для игрока, которому она досталась, то есть, не надо давать противнику возможности получить данную позицию. Получается, не надо получать позиции  $(7,34)$  и  $(8,33)$ . Поэтому позиция  $(7,33)$  проигрышная для Пети: удваивать ни одну кучу нельзя, Ваня удвоит большую кучу и выиграет, Добавить камень в одну из куч тоже нельзя, Ваня добавит камень во вторую и получит позицию  $(8,34)$ , в данном случае она будет проигрышной для Пети. Рассмотрим  $S = 32$ . Удваивать нельзя, Ваня увеличит вторую кучу вдвое и выиграет. Если же прибавить один камень, ситуация будет следующая: Ваня добавит камень в другую кучу, получит позицию  $(8,33)$ , и Петя своим вторым ходом не выиграет, максимум он получит  $(8,66)$ , что в сумме меньше  $77$ .



Получается, значений  $S$ , при которых Петя может выиграть в два хода, если добавит камень в одну из куч, больше нет. Другая стратегия – Петя будет своим первым ходом увеличивать в два раза одну из куч. Попробуем первую. После первого хода Пети позиция будет  $(14, S)$ , причем должны выполняться условия:

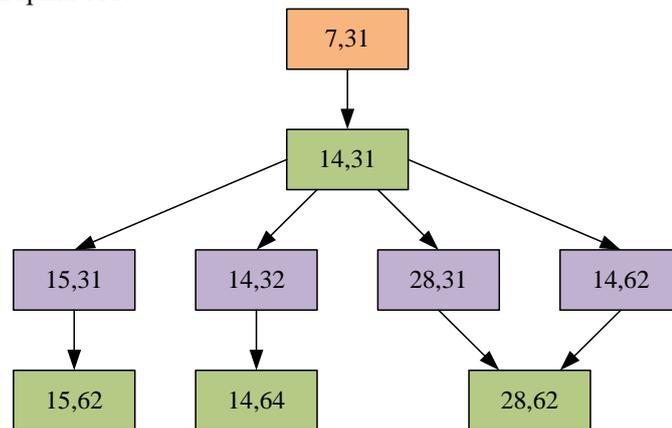
- Эта позиция не выигрышная, то есть,  

$$14 + 2S < 77 \Rightarrow S < 32;$$
- После добавления камня в любую кучу позиция становится выигрышной, то есть  

$$15 + 2S \geq 77 \Rightarrow S \geq 31$$
  

$$14 + 2(S + 1) \geq 77 \Rightarrow S \geq 31$$

Подходит  $S = 31$ , проверим это.



Если бы такое число было не найдено, можно было бы найти значения  $S$ , при которых Петя должен первым ходом удвоить число камней во второй куче. Это не 18 (показано в предыдущей задаче) и не 17 (получается  $(7,34)$ ), это выигрышная позиция, а она достанется Ване).

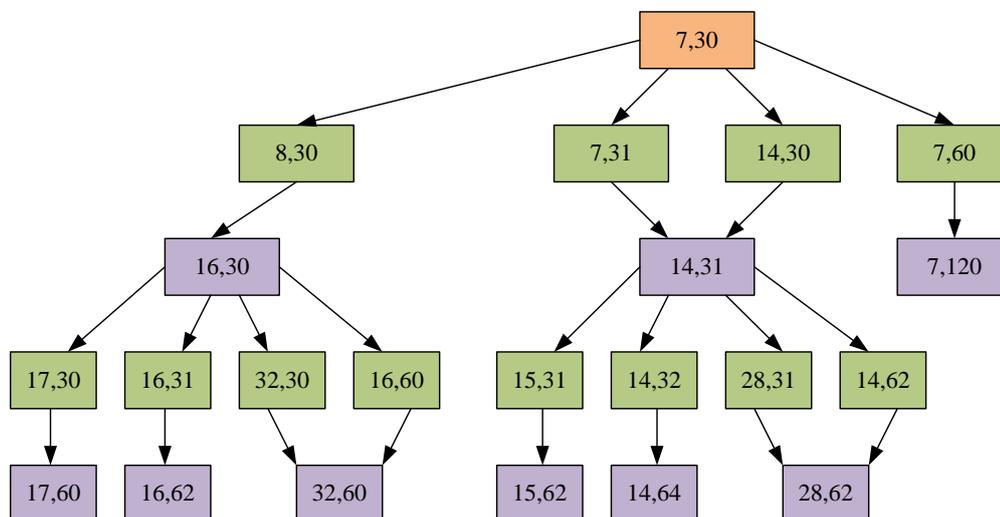
**Ответ. 31 34.**

### Задание 21. Выигрышная стратегия. Задание 3

Для игры, описанной в задании 19, найдите минимальное значение  $S$ , при котором одновременно выполняются два условия:

- у Вани есть выигрышная стратегия, позволяющая ему выиграть первым или вторым ходом при любой игре Пети;
- у Вани нет стратегии, которая позволит ему гарантированно выиграть первым ходом.

**Решение.** В предыдущей задаче показано, что при  $S = 31$  выигрывает Петя, а при  $S = 30$  нет (из условий). Проверяем  $S = 30$  для Вани (можно  $S = 33$ , но нужно наименьшее). На этот раз Петя делает все ходы, Ваня – только выигрышные. Учитываем, что позиция  $(14,31)$  проигрышная для того, кому она досталась, как показано в предыдущем пункте, то есть Ваня получает эту позицию из  $(7,31)$  и  $(14,30)$  и выигрывает. Из  $(8,30)$  Ваня получает  $(16,30)$ , если он будет прибавлять один камень, то Петя тоже будет прибавлять один камень, и Ваня, возможно, выиграет, но не в два хода. Эта стратегия Вани не подойдет для  $S = 29$ , поскольку, если Петя всегда прибавляет в меньшую кучу один камень, в результате Ваня получает ровно 77 камней на втором ходу (самая левая ветка дерева).



Возможно, Ваня выиграет при другой стратегии для меньшего значения  $S$ , если первым ходом удвоит не первую, а вторую кучу камней. Рассмотрим игру, аналогичную левой ветке дерева, когда Петя минимально увеличивает число камней, добавляя камень в меньшую кучу.

Начало:  $(7, S)$   
 Петя:  $(8, S)$   
 Ваня:  $(8, 2S)$   
 Петя:  $(9, 2S)$   
 Ваня:  $(9, 4S)$

$S$  должно удовлетворять условиям:

$$9 + 4S \geq 77 \Rightarrow S \geq 17$$

$$8 + 4S < 77 \Rightarrow S < 18$$

Но  $S = 17$  все же не удовлетворяет всем условиям, у Пети может быть другая стратегия: поставить  $(14,17)$ . Если Ваня удвоит 17, Петя затем удвоит 34 и выигрывает. В остальных случаях он не выиграет в два хода, например, из  $(28,17)$  Петя получает  $(28,18)$ , и Ване остается один ход, но этого недостаточно.

**Ответ. 30.**

## Задание 22. Анализ программы с циклами и условными операторами

Ниже на четырёх языках программирования записан алгоритм. Получив на вход число  $x$ , этот алгоритм печатает два числа:  $L$  и  $M$ . Укажите наибольшее число  $x$ , при вводе которого алгоритм печатает сначала 4, а потом 5.

C++	Python
<pre>#include &lt;iostream&gt; using namespace std;  int main() {     int x, L, M, Q;     cin &gt;&gt; x;     Q = 9;     L = 0;     while (x &gt;= Q) {         L = L + 1;         x = x - Q;     }     M = x;     if (M &lt; L) {         M = L;         L = x;     }     cout &lt;&lt; L &lt;&lt; endl &lt;&lt; M &lt;&lt; endl;     return 0; }</pre>	<pre>x = int(input()) Q = 9 L = 0 while x &gt;= Q:     L = L + 1     x = x - Q M = x if M &lt; L:     M = L     L = x print(L) print(M)</pre>
Алгоритмический язык	Паскаль
<pre>алг нач     цел  x, L, M, Q     ввод x Q := 9 L := 0 нц  пока  x &gt;= Q L := L + 1 x := x - Q кц M := x     если  M &lt; L         то M := L L := x     все     вывод L,  M кон</pre>	<pre>var x, L, M, Q: integer; begin     readln(x);     Q := 9;     L := 0;     while x &gt;= Q do         begin             L := L + 1;             x := x - Q;         end;     M := x;     if M &lt; L then         begin             M := L;             L := x;         end;     writeln(L);     writeln(M); end.</pre>

**Решение.** Начнем с конца. Рассмотрим следующие строки:

```
M = x;
if (M < L) {
    M = L;
    L = x;
}
```

Здесь по сути происходит следующее: в переменных  $M$  и  $L$  теперь хранятся значения  $L$  и  $x$ , какими они стали после выхода из цикла, причем  $L \leq M$ . Получается, что после выхода из цикла возможны два варианта:  $L = 4, x = 5$  и  $L = 5, x = 4$ .

Проанализируем цикл.

```
while (x >= Q) {
    L = L + 1;
    x = x - Q;
}
```

Переменная  $L$  представляет собой счетчик, сколько раз заходили в цикл, а от  $x$  в цикле каждый раз отнимается 9; т.е. после выхода из цикла  $x = x - 9L$ . Получается, чтобы получить максимальное значение  $x$ , которое было введено, надо взять вариант  $L = 5, x = 4$ ; тогда начальное значение  $x$  находится из условия:

$$4 = x - 9 \times 5 \Rightarrow x = 49.$$

Для проверки можно рассмотреть работу программы при  $x = 50$ . По выходе из цикла имеем  $L = 5, x = 5$ .

**Ответ. 49.**

### Задание 23. Оператор присваивания и ветвления. Перебор вариантов, построение дерева

Исполнитель преобразует число на экране.

У исполнителя есть две команды, которым присвоены номера:

1. Прибавить 1
2. Умножить на 2

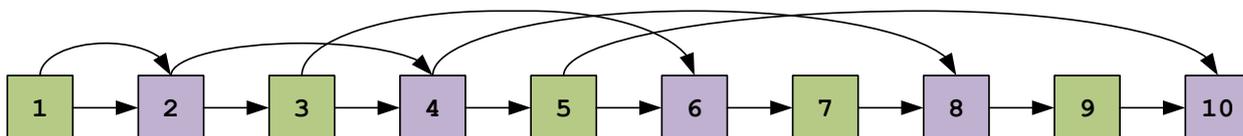
Первая команда увеличивает число на экране на 1, вторая умножает его на 2.

Программа для исполнителя – это последовательность команд.

Сколько существует программ, для которых при исходном числе 1 результатом является число 20, и при этом траектория вычислений содержит число 10?

Траектория вычислений программы – это последовательность результатов выполнения всех команд программы. Например, для программы 121 при исходном числе 7 траектория будет состоять из чисел 8, 16, 17.

**Решение.** Сначала найдем количество программ, которые переводят 1 в 10.



Если число  $k$  не кратно 2, оно получается только из  $k - 1$ ; числа, кратные 2, получаются двумя способами (еще и из  $k/2$ ). Составляем систему уравнений (тут можно идти от меньшего к большему).

$$\begin{aligned}N_1 &= 1 \\N_2 &= 2N_1 = 2 \\N_3 &= N_2 = 2 \\N_4 &= N_2 + N_3 = 4 \\N_5 &= N_4 = 4 \\N_6 &= N_3 + N_5 = 6 \\N_7 &= N_6 = 6 \\N_8 &= N_4 + N_7 = 10 \\N_9 &= N_8 = 10 \\N_{10} &= N_5 + N_9 = 14\end{aligned}$$

При переходе от 10 к 20 возможно только два варианта: либо 20 получается из 10 умножением на 2, либо от 10 к 20 добираться последовательным прибавлением единицы (если любое число, кроме 10, умножить на 2, получим число, большее 20). Итого, добраться от 1 до 10 можно 14 способами, и для каждого из этих способов есть два способа добраться до 20; имеем 28 вариантов.

**Ответ. 28.**

## Задание 24. Обработка символьных строк

Текстовый файл состоит не более чем из  $10^6$  символов X, Y и Z.

Определите максимальное количество идущих подряд символов, среди которых каждые два соседних различны. Для выполнения этого задания следует написать программу.

**Решение.** Проверяем два соседних символа, считанных из файла. Если они не равны, увеличиваем длину текущей цепочки на 1 (первоначально она равна 1, цепочка из одного символа удовлетворяет условиям). Сравниваем текущую длину цепочки с максимальной длиной, если она больше максимальной, обновляем максимальную. Сдвигаем символы: заменяем предыдущий текущим, текущий считываем из файла.

Всегда проверяйте, как работает программа, если искомая цепочка находится в конце файла, это особый случай.

**Пример.** XYXYXYXXY – содержимое файла

- $a, b$  – символы,  $a$  предыдущий,  $b$  текущий
- $curr\_len$  – текущая длина последовательности
- $max\_len$  – максимальная длина последовательности

a	b	curr len	max len	остаток файла	примечание
		1	1	XYXYXYXXY	
X	Y	2	2	YXYXYXXY	обновляем max_len
Y	X	1	2	XYXYXXY	a=b
Y	X	2	2	YXYXYXXY	
X	Y	3	3	XYXXY	обновляем max_len
Y	X	4	4	YXXY	обновляем max_len
X	Y	5	5	XXY	обновляем max_len
Y	X	6	6	XY	обновляем max_len
X	X	1	6	Y	a=b
X	Y	2	6		

Pascal	C++	Python
<pre> var curr_len, max_len: integer; a, b: char; f: text; begin   assign(f, '24_demo.txt');   reset(f);   read(f,a);   read(f,b);   curr_len := 1;   max_len := 0;   while not Eof(f) do begin     if (a &lt;&gt; b) then begin       curr_len:=curr_len+1;       if(curr_len&gt;max_len) then         max_len:=curr_len;     end     else curr_len:=1;     a:=b;     read(f,b);   end;   if (a &lt;&gt; b) then begin     curr_len:=curr_len+1;     if(curr_len&gt;max_len) then       max_len:=curr_len;   end;   writeln(max_len); end. </pre>	<pre> #include&lt;iostream&gt; #include&lt;fstream&gt;  using namespace std;  int main() {   ifstream f;   f.open("24_demo.txt");   int max_len=1, curr_len=1;   char a,b;   f&gt;&gt;a;   while (f&gt;&gt;b)   {     if (a!=b)     {       curr_len++;       if(curr_len&gt; max_len)         max_len = curr_len;     }     else       curr_len = 1;     a=b;   }   cout &lt;&lt; max_len;   f.close(); } </pre>	<pre> f=open('24_demo.txt','r') max_len = 1 curr_len = 1 a=f.read(1) b=f.read(1) while b:   if a!=b:     curr_len += 1     if curr_len&gt;max_len:       max_len=curr_len   else:     if curr_len&gt;max_len:       max_len=curr_len     curr_len = 1   a=b   b=f.read(1) print(max_len) f.close() </pre>

Решение задачи на Pascal <https://inf-ege.sdangia.ru/problem?id=27421>

Ответ. 35

## Задание 25. Обработка целочисленной информации

Напишите программу, которая ищет среди целых чисел, принадлежащих числовому отрезку [174457; 174505], числа, имеющие ровно два различных натуральных делителя, не считая единицы и самого числа. Для каждого найденного числа запишите эти два делителя в таблицу на экране с новой строки в порядке возрастания произведения этих двух делителей. Делители в строке таблицы также должны следовать в порядке возрастания.

Например, в диапазоне [5;9] ровно два целых различных натуральных делителя имеют числа 6 и 8, поэтому для этого диапазона таблица на экране должна содержать следующие значения:

```
2    3
2    4
```

**Решение.** Если делителей всего два, то один меньше корня, другой больше. При этом число не будет полным квадратом, поскольку у полного квадрата число различных делителей нечетно. Произведение этих делителей равно самому числу, поэтому просто перебираем числа в порядке возрастания. Начиная с  $j=2$ , проверяем, является ли  $j$  делителем  $n$ . Если да, увеличиваем счетчик делителей  $c$  и запоминаем делитель  $j$  в переменной  $m$ . Продолжаем проверки до тех пор, пока  $c < 2$  и  $j * j \leq n$ . У числа ровно два делителя, если мы нашли один (не превышающий корня из числа), при этом он не равен корню.

Pascal	C++	Python
<pre>var c,j,n,m:longint; begin for n:=174457 to 174505 do begin c:=0; j:=2; while((c&lt;2)and(j*j&lt;=n)) do begin if (n mod j = 0) then begin c:=c+1; m:=j; end; j:=j+1; end; if((c=1)and(m*m&lt;n)) then writeln(m, ' ',n div m); end end.</pre>	<pre>#include &lt;iostream&gt; using namespace std; int main() { long n,j,c,m; for (n=174457;n&lt;=174505;n++) { c=0; j=2; while((c&lt;2)&amp;&amp;(j*j &lt;=n)) { if (n%j==0) { c+=1; m=j; } j+=1; } if ((c==1) &amp;&amp; (m*m&lt;n)) cout&lt;&lt;m&lt;&lt;' '&lt;&lt;n/m&lt;&lt;endl; } }</pre>	<pre>for n in range(174457,174506): c=0 j=2 while c&lt;2 and j*j &lt;=n: if n%j==0: c+=1 m=j j+=1 if c==1 and m*m&lt;n: print(m,n//m)</pre>

**Ответ.**

**3 58153**

**7 24923**

**59 2957**

**13 13421**

**149 1171**

**5 34897**

**211 827**

**2 87251**

## Задание 26. Обработка целочисленной информации

Системный администратор раз в неделю создаёт архив пользовательских файлов. Однако объём диска, куда он помещает архив, может быть меньше, чем суммарный объём архивируемых файлов. Известно, какой объём занимает файл каждого пользователя.

По заданной информации об объёме файлов пользователей и свободном объёме на архивном диске определите максимальное число пользователей, чьи файлы можно сохранить в архиве, а также максимальный размер имеющегося файла, который может быть сохранён в архиве, при условии, что сохранены файлы максимально возможного числа пользователей.

В первой строке входного файла находятся два числа:  $S$  — размер свободного места на диске (натуральное число, не превышающее 10 000) и  $N$  — количество пользователей (натуральное число, не превышающее 1000). В следующих  $N$  строках находятся значения объёмов файлов каждого пользователя (все числа натуральные, не превышающие 100), каждое в отдельной строке.

Запишите в ответе два числа: сначала наибольшее число пользователей, чьи файлы могут быть помещены в архив, затем максимальный размер имеющегося файла, который может быть сохранён в архиве, при условии, что сохранены файлы максимально возможного числа пользователей.

*Пример входного файла:*

```
100 4
80
30
50
40
```

При таких исходных данных можно сохранить файлы максимум двух пользователей. Возможные объёмы этих двух файлов 30 и 40, 30 и 50 или 40 и 50. Наибольший объём файла из перечисленных пар — 50, поэтому ответ для приведённого примера:

```
2 50
```

**Решение.** Чем меньше файл занимает места на диске, тем больше остается свободного места и тем больше файлов можно туда поместить. Поэтому решение состоит из следующих шагов.

- Формируем массив  $a[N]$  из всех объёмов файлов и сортируем его любым алгоритмом.
- Накапливаем сумму  $sum$  элементов массива, перебирая их от меньшего к большему, пока она не превысит место на диске  $S$ , либо мы не просуммируем все  $N$  чисел. Первая часть ответа — количество чисел в наибольшей сумме, не превышающей  $S$ . Если  $sum + a[i]$  превышает  $S$ ,  $a[i]$  не добавляем.
- Пробуем заменить последнее добавленное число на большее, перебирая все числа, которые не вошли в сумму, в порядке возрастания. Останавливаемся, когда измененная сумма превышает размер свободного места, или когда выйдем за границу массива. Вторая часть ответа — последнее число, которое удалось добавить к сумме, и она не превысила  $S$ .

C++	Python
<pre> #include&lt;iostream&gt; #include&lt;fstream&gt; using namespace std; int main() {     ifstream F;     F.open("26_demo.txt");     int s,n,i,j,x;     F&gt;&gt;s&gt;&gt;n;     int a[n];     for (i=0;i&lt;n;i++)     {         F &gt;&gt; a[i];         j=i-1;         x=a[i];         while((j&gt;=0)&amp;&amp;(a[j]&gt;x))         {             a[j+1]=a[j];             j--;         }         a[j+1]=x;     }     int summa = 0, k = 0, Max = 0;     i=0;     while ((i&lt;n) &amp;&amp; (summa+a[i]&lt;=s))     {         summa=summa+a[i];         i++;     }     Max=a[i-1];     k=i;     summa=summa-Max;     while((i&lt;n) &amp;&amp; (summa+a[i]&lt;=s))     {         Max=a[i];         i+=1;     }     cout &lt;&lt; k &lt;&lt; " " &lt;&lt; Max;     return 0; } </pre>	<pre> f=open('26_demo.txt','r') s, n = map(int, f.readline().split()) a = [] for i in range(n):     a.append(int(f.readline())) a.sort() summa = 0 k = 0 Max = 0 i=0 while (i&lt;n) and (summa+a[i]&lt;=s):     summa=summa+a[i]     i+=1 Max=a[i-1] k=i summa=summa-Max while(i&lt;n) and (summa+a[i]&lt;=s):     Max=a[i]     i+=1 print(k,Max) </pre>

Решение задачи на Pascal <https://inf-ege.sdangia.ru/problem?id=27423>

**Ответ. 568 50**

## Задание 27. Программирование

Имеется набор данных, состоящий из пар положительных целых чисел. Необходимо выбрать из каждой пары ровно одно число так, чтобы сумма всех выбранных чисел не делилась на 3 и при этом была максимально возможной. Гарантируется, что искомую сумму получить можно.

Программа должна напечатать одно число – максимально возможную сумму, соответствующую условиям задачи.

*Входные данные.*

Даны два входных файла (файл А и файл В), каждый из которых содержит в первой строке количество пар  $N$  ( $1 \leq N \leq 100000$ ). Каждая из следующих  $N$  строк содержит два натуральных числа, не превышающих 10 000.

Пример организации исходных данных во входном файле:

```
6
1   3
5   12
6   9
5   4
3   3
1   1
```

Для указанных входных данных значение искомой суммы должно быть число 32.

В ответе укажите два числа: сначала значение искомой суммы для файла А, затем для файла В.

*Предупреждение:* для обработки файла В не следует использовать переборный алгоритм, вычисляющий сумму для всех возможных вариантов, поскольку написанная по такому алгоритму программа будет выполняться слишком долго.

**Решение.** В этой задаче эффективное решение проще неэффективного.

**Идея:**

- Максимальная сумма состоит из максимальных слагаемых
- Сказать, будет ли сумма кратна трем, нельзя, пока не посчитана вся сумма
- Будем составлять сумму из максимальных слагаемых, вдруг нам повезет
- Если не повезет, нам нужно минимально уменьшить сумму, для этого достаточно изменить свой выбор в одной паре. Мы уберем из суммы одно число и добавим другое, то есть, вычтем из суммы разность данных чисел. Это разность не должна быть кратной 3 и должна быть минимальной.

**Пример.**

- $a, b$  – текущие числа
- $sum$  – текущая сумма
- $d$  – минимальная разность между числами, не кратная 3

a	b	max(a, b)	sum	d= a-b	dmin	примечание	обновляем
			0		10000		
1	3	3	3	2	2	a-b <d,  a-b  mod 3 >0	dmin
5	12	12	15	7	2	a-b >d	
6	5	6	21	1	1	a-b <d,  a-b  mod 3 >0	dmin
5	4	5	26	1	1	a-b =d	
4	4	4	30	0	1	a-b <d,  a-b  mod 3 =0	

Здесь ответ  $sum - dmin = 30 - 1 = 29$ .

C++	Python
<pre> #include&lt;iostream&gt; #include&lt;fstream&gt; using namespace std; int main() {     ifstream f("A.txt");     long n, a, b, i, d, dmin=10000, sum=0;     f &gt;&gt; n;     for(i=0; i&lt;n; i++)     {         f &gt;&gt; a &gt;&gt; b;         if (a&gt;b)         {             sum+=a;             d=a-b;         }         else         {             sum+=b;             d=b-a;         }         if ((d%3!=0) &amp;&amp; (d&lt;dmin))             dmin=d;     }     if (sum%3==0)         sum-=dmin;     cout&lt;&lt; sum;     return 0; } </pre>	<pre> f=open('A.txt','r') n=int(f.readline()) sum=0 dmin=10000 for i in range(n):     a,b = map(int, f.readline().split())     if a&gt;b:         sum+=a         d=a-b     else:         sum+=b         d=b-a     if d%3!=0 and d&lt;dmin:         dmin=d if sum%3==0:     sum-=dmin print(sum) f.close() </pre>

Решение задачи на Pascal <https://inf-ege.sdamgia.ru/problem?id=27424>

**Ответ. 127127 399762080**