

Задания Интернет-олимпиады 2019-2020 для 9-10 классов

Тур 2

Задача 1.

Штирлиц, находясь уже на пенсии, на полях книги записал следующую шифровку:
“>сов Э и2:20 с.620”

Какое число, относящееся и к информатике, описано на закодированной странице?

Решение. Штирлиц имел в виду Большую Советскую энциклопедию, издание 2, том 20. Далее надо было найти скан соответствующего тома, например, здесь:

http://publ.lib.ru/ARCHIVES/S/Sovetskaya_Enciklopediya/Sovetskaya_Enciklopediya.html

При необходимости научиться работать с форматом djvu.

На 620й странице этого тома вводится понятие **кило**, что означает 1000.

Ответ. 1000

Задача 2.

По каналу связи передаются сообщения, содержащие только восемь букв: К, Л, М, Н, О, П, Р, С. Для передачи используется неравномерный двоичный код в котором никакой более короткий код не является началом более длинного кода. Кодовые слова для некоторых букв известны: К – 001, Н – 100, Р – 11 (каждой из остальных букв нужно назначить свой код).

Какое наименьшее количество двоичных знаков потребуется для кодирования слова МОЛОКОСОС?

Решение

Для решения этой задачи можно построить дерево кодов. Для начала необходимо заполнить дерево уже известными кодами. Затем для оставшихся букв необходимо посчитать их количество вхождений:

- О – 4
- С – 2
- М – 1
- К – 1 (уже есть код)
- Л – 1
- Н – 0 (уже есть код)
- П – 0
- Р – 0 (уже есть код)

Для оставшихся букв нужно построить коды. Но для букв, которые встречаются чаще, код должен быть короче.



- O – 4 (01)
- C – 2 (101)
- M – 1 (0001)
- K – 1 (001)
- Л – 1 (00001)
- H – 0 (100)
- П – 0 (00000)
- P – 0 (11)

Ответ считается как сумма длин кодов для каждой буквы слова.

Ответ. 26

Задача 3.

Программа вводит натуральное число x , выполняет преобразования, а затем выводит одно число. Укажите **наибольшее** возможное значение x , не превосходящее 2 000 000 000, при вводе которого программа выведет число 12.

Python	C++	Pascal
<pre>x = int(input()) a=0 b=10 while x > 0: d = x % 6 if d > a: a = d if d < b: b = d x = x // 6 print(a*b)</pre>	<pre>#include <iostream> using namespace std; int main() { int x, a, b, d; cin >> x; a = 0; b = 10; while (x > 0) { d = x % 6; if (d > a) a = d; if (d < b) b = d; x = x / 6; } cout << a*b << endl; return 0; }</pre>	<pre>var x, a, b, d: longint; begin readln(x); a := 0; b := 10; while x > 0 do begin d := x mod 6 if d > a then a := d; if d < b then b := d; x := x div 6 end; writeln(a*b) end.</pre>

Решение

Для решения этой задачи достаточно понять, что программа переводит исходное число в шестеричную систему счисления. В переменной a будет храниться максимальная цифра, а b будет храниться минимальная цифра числа в шестеричной системе счисления. Заметим, что 12 – это произведение 2 цифр. Значит a и b – делители числа 12. Все цифры в шестеричной системе счисления меньше 6. Единственная возможная комбинация – это числа 4 и 3. Максимальное число до 2 000 000 000 равно $44444444443_6 = 1741425867_{10}$

Ответ. 1741425867

Задача 4.

Найдите количество различных неориентированных связанных графов на четырех различных вершинах. Например, для трех вершин таких графов четыре.

Решение. Между четырьмя различными вершинами можно провести 6 ребер, поэтому общее число графов на четырех вершинах равно $2^6 = 64$. Подсчитаем число несвязных графов:

- 1) три вершины образуют связную компоненту (4-мя способами) + отдельностоящая вершина; всего таких графов $4 \times 4 = 16$;
- 2) две связные компоненты по две вершины – 3 варианта;
- 3) две вершины соединены, а две другие – нет: 6 вариантов;
- 4) ребер в графе нет – 1 вариант.

$$64 - (16 + 3 + 6 + 1) = 38.$$

Решение

Ответ. 38

Задача 5.

При хранении факториала некоторого натурального числа в $k > 37$ двоичных разрядах три старших бита были равны нулю, но факториал уже следующего числа сохранить в тех же k разрядах было невозможно. Позже было решено хранить в этих разрядах только значащие двоичные цифры факториала – без хвостовых нулей. Факториал какого максимального числа можно сохранить теперь. В случае, если правильных ответов несколько, то запишите ответ для минимально возможного значения k .

Решение. Так как при домножении на число l двоичное значение факториала увеличилось как минимум на 4 разряда, то l равно как минимум 11. При этом $14!$ занимает как раз 37 двоичных разрядов (выяснить это можно с помощью выражения на Python:

```
len(bin(14*13*12*11*10*9*8*7*6*5*4*3*2))-2
```

То есть изначально значение факториала было равно как минимум 14 (40 двоичных разрядов). При домножении на 15 двоичное значение увеличится на 4 разряда, что больше трёх. Значит, общее число разрядов для хранения было равно 40. Если в них не хранить двоичные нули, то сохранить можно все значащие цифры числа $18!$ в двоичном представлении которого 37 цифр, кроме хвостовых нулей. $19!$ Имеет уже 41 значащую

цифру.

Для иллюстрации можно запустить следующий код:

```
n = 1
for i in range(1, 25):
    n *= i
nb = bin(n)[2:]
nbz = nb[:nb.rfind('1') + 1]
print(i, len(nb), len(nbz), nb, nbz)
```

Ответ. 18