Всероссийская олимпиада школьников по информатике
Решения задач муниципального этапа, 7-8 кл., 5 декабря 2016 г.

A. Билеты

(Почти фольклор, решение задачи — *Киндер М.И.*)

Начальное значение суммы s считаем равным 0. В цикле просматриваем значения всех чисел, обозначающих возраст участников группы. Если возраст меньше 11, то значение суммы s не изменяется, и мы переходим к просмотру следующего числа. Если возраст меньше 19, значение суммы s увеличиваем на 50 (руб.), в противном случае увеличиваем s на 100 (руб.).

```
for i := 1 to n do  begin
    read(d);
    if d < 11 then continue
    if d < 19 then Inc(s, 50) else Inc(s, 50);
end;
write(s);
```

B. Палиндром

(Почти фольклор, решение задачи — *Киндер М.И.*)

В десятичной записи любого палиндрома равноудаленные от концов цифры совпадают. Из n цифр заданного набора можно будет составить палиндром только в том случае, когда каждая цифра от 0 до 9 входит в этот набор *чётное* количество раз (возможно, ни разу), или же *ровно одна* цифра входит нечётное число раз, а все остальные — чётное число. Например, последнему условию удовлетворяют числа 1112002 и 2200550, из которых можно составить палиндромы 2011102 и 2050502. В записи этих палиндромов цифра, входящая *нечётное* число раз, находится ровно посередине. Исключение составляют числа, которые содержат чётное число нулей и ровно одну ненулевую цифру. К ним относится, в частности, число 10000, из которого составить палиндром невозможно. Реализация этого алгоритма требует аккуратной проверки всех этих условий, а также умения работать с отдельными цифрами-символами исходного набора. Сложность этого алгоритма — $O(n)$ операций.

C. Сумма факториалов

(Автор задачи и решений — *Киндер М.И.*)

Первое решение. Воспользуемся жадным алгоритмом. Из данного числа x вычтем *наибольший* возможный факториал $n!$ и к полученной разности применим те же рассуждения. Другими словами, из полученной разности снова вычтем наибольший факториал, и так далее. Количество вычитаемых факториалов и будет искомым ответом в задаче. Реализация этого алгоритма несложная. Корректность жадного алгоритма следует из неравенства

$$1\cdot 1! + 2\cdot 2! + \ldots + (n-1)\cdot(n-1)! < n!\,.$$

Действительно, если для заданного числа x выполняется неравенство $n! \le x < (n+1)!$, то представление x в виде суммы наименьшего количества факториалов обязательно содержит слагаемое $n!$. В противном случае, как следует из приведённого неравенства, сумма «остальных» факториалов будет меньше x. (Если факториал $k!$ входит с коэффициентом $c_k > k$, то слагаемое $c_k \cdot k! = (k+1)! + (c_k - k - 1)k!$ можно заменить суммой меньшего числа слагаемых.)

Второе решение основано на применении *факториальной системы счисления*. В этой системе счисления основанием служит последовательность натуральных чисел: $1!, 2!, \ldots, k!$. В факториальной системе натуральное число x представляется единственным образом в виде

$$x = c_1 \cdot 1! + c_2 \cdot 2! + \ldots + c_n \cdot n!, \text{ где } 0 \le c_k \le k.$$

Из этого представления следует, что *наименьшее* количество факториалов, при сложении которых получается заданное x, равно сумме «цифр» c_k в записи числа x в факториальной системе счисления, то есть равно

$$s = c_1 + c_2 + \ldots + c_n\,.$$

Осталось научиться переводить число в факториальную систему. Оказывается, для этого не обязательно вычислять факториалы натуральных чисел. Действительно, каждое из чисел $k!$ делится на все натуральные числа от 1 до k, поэтому слагаемые вида $c_k \cdot k!$ делятся на 2 при любом $k \geq 2$. Значит, число c_1 равно остатку от деления x на 2. Разделив нацело x на 2, то есть отбрасывая слагаемое $c_1 \cdot 1!$, применим те же рассуждения к числу x div 2. Теперь число c_2 равно остатку от деления x div 2 на 3. Этот процесс продолжаем до тех пор, пока частное от деления на очередное число k не станет уже равным нулю. Ниже приведен фрагмент кода на языке Delphi, реализующего этот алгоритм.

```
s := 0;   k := 2;
while  (n > 0)  do  begin
      s := s + x mod k;
      x := x div k;
      inc(k);
end;
write(s);
```

D. НОК

(Легенда составлена студентами КФУ, автор задачи и решений — *Киндер М.И.*)

РЕШЕНИЕ ПОДЗАДАЧИ 1. Для «небольших» значений k и m ($2 \leq k \leq 3$, $1 \leq m \leq 100$) все требуемые наборы можно найти несложным перебором. Такое решение проходит тесты на 30 баллов.

РЕШЕНИЕ ПОДЗАДАЧИ 2. Поскольку требуется подсчитать только количество требуемых наборов чисел, можно применить комбинаторные рассуждения. Пусть m — наименьшее общее кратное всех чисел нашего набора, и пусть $m = p_1^{\alpha_1} p_2^{\alpha_2} \ldots p_r^{\alpha_r}$ — разложение m на простые множители. Тогда условию задачи удовлетворяют все наборы чисел $a_i = p_1^{\alpha[i,1]} p_2^{\alpha[i,2]} \ldots p_r^{\alpha[i,r]}$, $1 \leq i \leq k$, у которых неотрицательные показатели $\alpha[i,s]$ степеней простых чисел p_s не превосходят α_s и для которых выполняются условия

$$\max(\alpha[1,s], \alpha[2,s], \ldots, \alpha[k,s]) = \alpha_s,$$

для всех $1 \leq s \leq r$. Подсчитаем число наборов по каждому показателю α отдельно. Количество наборов, которые удовлетворяют условиям $0 \leq \alpha_i \leq \alpha$ для всех i от 1 до k, равно

$$(\alpha + 1)^k - \alpha^k.$$

В самом деле, число α_i может принимать $\alpha + 1$ значений — это все неотрицательные числа от 0 до α, и по правилу произведения количество таких наборов равно $(\alpha + 1)^k$. Осталось исключить из этого множества наборы, для которых $\max(\alpha_1, \alpha_2, \ldots, \alpha_k) < \alpha$. Для таких наборов $0 \leq \alpha_i \leq \alpha - 1$ для всех i от 1 до k, поэтому их количество равно α^k. Значит, количество требуемых наборов равно $(\alpha + 1)^k - \alpha^k$. Теперь по правилу произведения число наборов по *всем* показателям α_s равно:

$$L(m,k) = \prod_{i=1}^{r} \left[(\alpha_i + 1)^k - \alpha_i^k\right].$$

Например, для $m = 10 = 2^1 \cdot 5^1$ и $k = 2$ получаем

$$L(10,2) = [(1+1)^2 - 1^2] \cdot [(1+1)^2 - 1^2] = 9$$

упорядоченных наборов из двух чисел, у которых НОК равен 10. Алгоритмическая сложность такого решения — $O(Fact(m) + k)$.Такое решение проходит тесты ещё на 35 баллов.

РЕШЕНИЕ ПОДЗАДАЧИ 3. Для чисел k из диапазона $[1; 10^{18}]$ для быстрого вычисления $(\alpha + 1)^k$ и α^k используем бинарный алгоритм возведения в степень. Алгоритмическая сложность такого решения — $O(Fact(m) + \log k)$.

Председатель жюри М.И. Киндер