

**Муниципальный этап**  
**Всероссийской олимпиады школьников**  
**по информатике и ИКТ**  
**9-11 классы**  
**2018 - 2019 учебный год**

**Задача А**

Обозначим искомое количество пенсионеров  $X$ . Тогда должно выполняться соотношение:

$$\frac{K + X}{N + X} \geq \frac{1}{3}$$

Выражая  $X$ , получаем:

$$X \geq \frac{N - 3K}{2}$$

Т.е.  $X$  равен выражению справа, если результат деления - целое число, или целой части выражения, увеличенной на 1, если результат нецелое. В программе это можно вычислить, пользуясь приемом вычисления частного с округлением вверх, либо используя команду ветвления.

Кроме того, следует проверить, не было ли достаточным количество пенсионеров изначально, и в этом случае вывести 0.

**Пример решения на языке PascalABC**

```
program sovet;  
var n, k, x:integer;  
begin  
  read(n);  
  read(k);  
  x := (n - 3 * k + 1) div 2;  
  if x < 0 then x := 0;  
  writeln(0);  
end.
```

## Задача В

У Миши есть три варианта действий:

- подняться по лестнице

В этом случае, ему придется подниматься  $n - 1$  этажей, затрачивая  $a$  секунд на каждый:

$$t1 = (n - 1) * a;$$

- дождаться, пока лифт спустится до 1-го этажа, а затем подняться на нужный этаж;

Лифт будет спускаться  $k - 1$  этажей, а затем подниматься  $n - 1$  этажей, затрачивая  $b$  секунд на каждый:

$$t2 = (k - 1) * b + (n - 1) * b;$$

- подняться по лифту до некоторого этажа  $X$ , пока лифт до него спускается, а затем на лифте подняться до нужного этажа.

Найдем время, которое Миша может подниматься вверх, пока лифт спускается.

Скорость Миши  $\frac{1}{a}$ , скорость лифта -  $\frac{1}{b}$ , тогда их скорость сближения  $\frac{a+b}{ab}$ , а

первоначальное расстояние между ними  $k-1$ . Значит, доступное время для

подъема Миши  $t = \frac{(k-1) \cdot a \cdot b}{a+b}$  (целая часть). За это время Миша с первого этажа

может подняться до этажа  $X = 1 + \left\lceil \frac{t}{a} \right\rceil$  и там дождаться лифта. Значит, лифт

должен спуститься до этажа  $X$ , а затем подняться до этажа  $N$ :

$$t3 = (k - x) * b + (n - x) * b;$$

Среди всех вычисленных значений времени, следует выбрать минимальное.

## Пример решения на языке PascalABC

```
program lift;
var a,b,k,n,t,x, t1, t2, t3:integer;
begin
read(n,k,a,b);
t1 := (n - 1) * a;
t2 := (k - 1) * b + (n - 1) * b;
t := ((k - 1) * a * b) div (a + b);
x := 1 + t div a;
t3 := (k - x) * b + (n - x) * b;
write(min (t1, min (t2, t3)));
end.
```

### Задача С

Изобразим плиточный пол в виде квадратной таблице (в программе для нее потребуется двумерный массив).

Рассмотрим процесс нагревания пола при одном источнике тепла. Звездочкой обозначим источник тепла, а числами - номер хода, на котором данная плитка прогреется:

7	6	5	4	5	6	7	8	9	10
6	5	4	3	4	5	6	7	8	9
5	4	3	2	3	4	5	6	7	8
4	3	2	1	2	3	4	5	6	7
3	2	1	*	1	2	3	4	5	6
4	3	2	1	2	3	4	5	6	7
5	4	3	2	3	4	5	6	7	8
6	5	4	3	4	5	6	7	8	9
7	6	5	4	5	6	7	8	9	10
8	7	6	5	6	7	8	9	10	11

Заметим, что если у источника тепла координаты  $X, Y$ , то определить время, когда клетка  $a[i, j]$  будет нагрета, можно по формуле

$$a[i, j] = |X - i| + |Y - j|$$

При этом источников тепла может быть несколько, поэтому нужно для каждой плитки определять её время нагревания каждым источником, выбирая минимальное из них.

Ответом к задаче будет максимальное время, полученное в таблице.

## Пример решения на языке PascalABC

```
program pol;  
var  
v:integer;  
  i,j,k,n,max,min:integer;  
  x,y:array[1..10] of integer;  
begin  
  readln(n,v);  
  for i := 1 to v do  
    readln(x[i],y[i]);  
    max := 0;  
  for i := 1 to n do  
    for j := 1 to n do  
      begin  
        min := 2000;  
        for k := 1 to v do  
          if abs(i - x[k]) + abs(j - y[k]) < min then  
            min:=abs(i - x[k]) + abs(j - y[k]);  
          if max < min then  
            max := min;  
        end;  
      end;  
  write(max);  
end.
```

## Задача D

Выполним действия Шамана в обратном порядке.

Известно, что в конце была получена последовательность из первых  $N$  заглавных букв английского алфавита. Сгенерируем нужную строку  $s$ , а затем будем составлять ее символы в нужном порядке.

Очевидно, что тот символ, с которого Шаман начинал складывать новую стопку на магическом круге, в получившейся последовательности  $s$  будет располагаться в середине, с него и нужно начинать. В дальнейшем нужно от центрального символа брать карточки, лежащие перед ним и после него. Также следует учитывать, что при нечетном  $N$  количество букв слева и справа от центрального символа будет одинаковым, а в случае четного  $N$  – символов справа от центрального будет на один меньше, чем слева.

### Пример решения на языке PascalABC

```
program cards;
var N, i, k:integer; s, a: string; c: char;
begin
  read(N); c := 'A'; s:= '';
  for i := 1 to N do
    begin
      s := s + c;
      c := succ(c);
    end;
  k := n div 2 + 1;
  a := s[k];
  for i := 1 to n div 2 do
    if (n mod 2 = 0) and (i = n div 2) then
      a := a + s[k - i]
    else
      a := a + s[k - i] + s[k + i];
  writeln(a);
end.
```

## Задача E

Очевидно, что переборное решение выйдет за допустимые пределы времени.

Проведём более подробный анализ представленной последовательности

2, 4, 6, 9, 12, 15, 20, 25, 30, 35, 42, 49, ... .

Из начального члена два следующих получаются добавлением 2 — наименьшего простого числа. Потом три числа получаются добавлением 3 — следующего простого числа. А вот следующие четыре числа получаются добавлением 5 — очередного простого числа. Просмотрев ещё несколько членов последовательности, замечаем, что количество добавлений очередного простого числа равно разности следующего простого числа и предыдущего простого числа. Так 3 добавляется  $(5 - 2)$  раз, 5 —  $(7 - 3)$  раз, 7 —  $(11 - 5)$  раз и так далее. Это позволяет не вычислять все члены последовательности подряд, а пропускать их довольно значительную часть.

### Пример решения на языке PascalABC

```
program seq;
var n, p1, p2, x, q : longint; t : boolean;
begin
  read(n);
  if n < 4 then
    write(2 * n)
  else
    begin
      p1:=3; p2:=5;
      while p1 + p2 - 2 < n do
        begin
          t := false;
          x := p2;
          while not t do
            begin
              t := true;
              q := 3;
              x := x + 2;
              while t and (q * q <= x) do
                begin
                  t := (x mod q <> 0);
```

```
        q := q + 2;  
    end;  
end;  
p1 := p2; p2 := x;  
end;  
write( p1 * (n - p1 + 2));  
end;  
end.
```