

## Задача А. Перерыв на лекции

Задача является арифметической.

Исходя из условия, следует вычислить время середины лекции и «отложить» в обе стороны от середины время, равное половине перерыва.

Если длительность лекции или длительность перерыва не делятся нацело на два, то следует учесть, что длительность первой части должна быть больше.

### Пример программы на языке Pascal:

```
var
  a, b, c, d, m, start, fin, dlit, middle, beg: integer;

begin
  read(a, b, c, d, m);
  start := 60 * a + b;
  fin := 60 * c + d;
  dlit := fin - start;
  if dlit mod 2 = 0 then
    begin
      middle := start + dlit div 2;
      beg := middle - m div 2;
    end
  else
    if m mod 2 <> 0 then
      beg := start + (dlit - m) div 2
    else
      begin
        middle := start + (dlit + 1) div 2;
        beg := middle - m div 2;
      end;
    write(beg div 60, ' ', beg mod 60)
  end.
```

## Задача В. Роботы

Для решения задачи необходимо посчитать, сколько раз робот А и робот В оказались бы одновременно в одной клетке, если бы оба начали движение одновременно из одной точки (например, будем рассматривать ситуацию, что оба двигались бы из левого верхнего в правый нижний угол).

Для того, чтобы роботы оказались в одной клетке, требуется, чтобы к одному и тому же моменту времени они сделали одинаковое количество шагов вниз и одинаковое количество вправо, причем неважно в каком порядке.

Будем для каждого робота считать количество шагов, сделанных вправо и вниз, и в каждый момент времени сравнивать, совпадают ли эти величины.

Для того, чтобы сравнивать траектории роботов, «инвертируем» траекторию робота В: будем рассматривать его движение из левого верхнего угла в правый нижний (т.е. команды будем смотреть с конца), и команды «L» и «U» заменим соответственно на «R» и «D».

### Пример программы на языке Pascal:

```
var
  n, m, count, rightA, downA, rightB, downB, i:
  integer;
  a, b, c, d: string;

begin
  readln(m);
  readln(n);
  readln(a);
  read(b);
  rightA := 0;
  downA := 0;

  rightB := 0;
  downB := 0;

  count := 0;

  for i := 1 to length(a) do
  begin
    if a[i] = 'R' then
      rightA := rightA + 1
    else
      downA := downA + 1;

    if b[length(b) + 1 - i] = 'L' then
      rightB := rightB + 1
    else
      downB := downB + 1;

    if (downA = downB) and (rightA = rightB) then
      count := count + 1;
  end;
  write(count + 1);
end.
```

### Задача С. Считалочка

С учетом имеющихся ограничений на  $N$  и  $K$  данная задача может быть решена моделированием данного процесса.

Создадим массив, в котором будем хранить 1, если участник с соответствующим номером еще не выбыл, и 0 – в противном случае.

Далее будем вычислять, кто из участников будет удален на следующем круге считалки, и заменять соответствующий элемент нулем (о тех пор, пока в массиве не остается единственный ненулевой элемент).

#### Пример программы на языке Pascal

```
var
  n, k, i, j, d: integer;
  a: array[0..1000] of byte;

begin
  read(n, k);
  for i := 0 to n - 1 do
    a[i] := 1;
  d := 0;
  i := 0;
  j := 0;
  while d <> n - 1 do
  begin
    if a[i] = 1 then
    begin
      j := j + 1;
      if j = k then
      begin
        a[i] := 0;
        j := 0;
        d := d + 1;
      end;
    end;
    i := (i + 1) mod n;
  end;
  i := 0;
  while a[i] = 0 do
    i := i + 1;
  write(i + 1);

end.
```

### Задача D. Блинный конвейер

Рассмотрим, как будут переворачиваться блинчики для  $n = 10$ .

Представим сковороды с блинами в виде последовательности натуральных чисел.

Зеленым цветом обозначим «блины», выпекающиеся с первой стороны, а красным – те, которые обжариваются уже со второй стороны.

Минута	Блины
1	1 2 3 4 5 6 7 8 9 10
2	1 2 3 4 5 6 7 8 9 10
3	1 2 3 4 5 6 7 8 9 10
4	1 2 3 4 5 6 7 8 9 10
5	1 2 3 4 5 6 7 8 9 10
6	1 2 3 4 5 6 7 8 9 10
7	1 2 3 4 5 6 7 8 9 10
8	1 2 3 4 5 6 7 8 9 10
9	1 2 3 4 5 6 7 8 9 10
10	1 2 3 4 5 6 7 8 9 10

В результате – с одной стороны в данный момент обжарено три блина – с номерами 1, 4 и 9.

Можно заметить, что в результате такого алгоритма «переворачиваются» блины каждый раз, когда идет минута с номером, являющимся делителем данного числа.

Для того, чтобы блин обжарился с одной стороны, требуется, чтобы он был перевернут нечетное количество раз. А значит, у номера данного блина должно быть нечетное количество делителей, т.е. номер является полным квадратом натурального числа.

Поэтому задача сводится к поиску количества полных квадратов, попадающих в промежуток  $[1; n]$ .

### Пример программы на языке Pascal

```
var n : longint;  
i: integer;  
begin  
  read(n);  
  
  i := 1;  
  while i * i <= n do  
    i := i + 1;  
  write(i - 1);  
end.
```

### Задача Е. Шкафчики в раздевалке

Если сумма чисел на шкафчике меньше задуманного Жорой числа, то Жоре будет достаточно заменить одну табличку на большое число, чтобы «добрать» нужную величину.

Если сумма чисел равна задуманному числу, то ни одну табличку менять не нужно.

Если же сумма больше задуманного числа, то нужно найти минимальное количество табличек так, чтобы сумма чисел на этих табличках давала разницу между задуманным числом и суммой чисел на шкафчиках. При переборе разных вариантов программа может работать долго, поэтому одним из способов поиска является бинарный поиск (по ответу).

При реализации программы следует учитывать, что диапазон значений числа  $S$  ( $1 \leq S \leq 10^{18}$ ) и выбирать подходящий тип данных.

### Пример программы на языке Pascal

```
var
  n, s, summa, left, right, mid: uint64;

begin
  read(n, s);
  summa := n * (n + 1) div 2;
  if s > summa then
    write('1')
  else
    if s = summa then
      write('0')
    else
      begin
        left := -1;
        right := 1000000000 + 100;
        while left + 1 < right do
          begin
            mid := (left + right) div 2;
            if n - mid + mid * (mid + 1) div 2 <= s then
              left := mid
            else
              right := mid;
          end;
          write(n - left);
        end
      end
end.
```