

Разбор задач

30 Районная олимпиада школьников Красноярского края по информатике, 7-8 классы

Основной тур от 1 декабря 2016 г.

ID	Задача	Тема	Балл	%
1431	А. Пираты	Задачи для начинающих	100	16
1429	В. Лягушка	Задачи для начинающих	100	14
1421	С. Число	Задачи для начинающих	100	18
1426	Д. Снежинка	Задачи для начинающих	100	22
1425	Е. Правильная таблица	Моделирование	100	52

Задача А. Пираты

(Время: 1 сек. Память: 16 Мб Баллы: 100)

В ожесточённом бою сражалось N пиратов. В результате A пиратов потеряли один глаз, B пиратов – одно ухо, C пиратов – одну руку и D пиратов остались без одной ноги.

Страховую компанию «Весёлый Роджер», где были застрахованы все пираты по тарифу «All inclusive», интересует: какое минимально возможное число пиратов могло потерять глаз, ухо, руку и ногу одновременно?

Требуется написать программу, определяющую величину, которая интересует страховую компанию.

Входные данные

Входной файл INPUT.TXT содержит 5 целых чисел: N , A , B , C и D – величины, описанные в условии задачи. Ограничения: $1 \leq N \leq 10^{18}$, $0 \leq A, B, C, D \leq N$.

Выходные данные

В выходной файл OUTPUT.TXT выведите целое число – ответ на задачу.

Примеры

№	INPUT.TXT	OUTPUT.TXT
1	100 70 75 80 90	15
2	100 100 100 100 100	100

Задача А. Пираты

N - всего пиратов

A - пиратов без глаза

B - пиратов без уха

C - пиратов без руки

D - пиратов без ноги

Минимальное количество пиратов, получивших все 4 увечья:

$$S = \max(0, A+B+C+D-3*N)$$



Чтобы получить как можно меньше пиратов со всеми четырьмя увечьями, нужно чтобы как можно меньше было общих пересечений по увечьям. Для чего расставим пиратов по кругу и начнем поочередно наделять увечьями так, чтобы каждое последующее продолжало предыдущее. Получим, что 4 увечья получают как минимум те, у кого на схеме присутствуют все 4 линии.

```
//Pascal
uses Math;

var n,a,b,c,d : int64;

begin
  read(n,a,b,c,d);
  write(max(0, a+b+c+d-3*n));
end.
```

```
//C++
#include <iostream>
#include <algorithm>

using namespace std;

long long n,a,b,c,d;

int main(){
  cin >> n >> a >> b >> c >> d;
  cout << max(0LL, a+b+c+d-3*n);
  return 0;
}
```

Задача В. Лягушка

(Время: 1 сек. Память: 16 Мб Баллы: 100)

В точке 0 на числовой прямой сидит лягушка. Она начинает прыгать вправо на 1 каждую секунду, пока не достигнет координаты K . После чего она начинает прыгать на 1 влево, пока не вернется в точку 0. Затем она снова прыгает вправо и т.д. Требуется определить, где окажется лягушка через T секунд.

Входные данные

Входной файл INPUT.TXT содержит натуральные числа K и T , не превосходящие 10^{18} .

Выходные данные

В выходной файл OUTPUT.TXT выведите одно число – координату лягушки в момент времени T .

Примеры

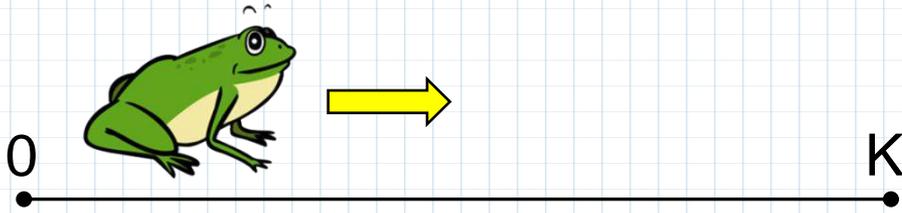
№	INPUT.TXT	OUTPUT.TXT
1	5 2	2
2	5 6	4

Система оценки

Решения, работающие только для $T \leq 10^6$, будут оцениваться в 40 баллов.

Решения, работающие только для $T \leq 10^9$, будут оцениваться в 80 баллов.

Задача В. Лягушка



K - длина отрезка, где прыгает лягушка
T - общее число ее прыжков (секунд)

Через $2K$ секунд лягушка возвращается в точку 0 и действие повторяется. Аналогично это происходит через $4K$, $6K$ и т.д. Поэтому период равен $2K$ и можно произвести замену:

$$T = T \bmod (2K)$$

Откуда получаем решение:

```
read(K, T)
T = T mod (2*K)
if (T < K) write(T)
else write(2*K - T)
```

```
//Pascal
var k,t : int64;

begin
  read(k,t);
  if odd(t div k) then
    write(k - t mod k)
  else
    write(t mod k)
end.

//C++
#include <iostream>

using namespace std;

long long k,t;

int main(){
  cin >> k >> t;

  if(t/k%2) cout << k - t%k;
  else cout << t%k;

  return 0;
}
```

Задача С. Число

(Время: 1 сек. Память: 16 Мб Баллы: 100)

Дано натуральное число. Требуется разделить запятыми тройки его цифр (считая справа).

Входные данные

Входной файл INPUT.TXT содержит натуральное число, не превосходящее 10^{100} .

Выходные данные

В выходной файл OUTPUT.TXT выведите то же число, разделяя тройки цифр запятыми.

Примеры

№	INPUT.TXT	OUTPUT.TXT
1	1000	1,000
2	123456789	123,456,789
3	12345	12,345
4	45	45

Задача С. Число

В силу ограничений на число ($N \leq 10^{100}$) во входных данных удобнее всего чтение производить в строковую переменную.

Далее, будем выводить данную строку посимвольно и вставлять символ "," каждый раз, когда оставшееся количество символов кратно трем (за исключением последнего символа):

```
string n
```

```
read(n)
```

```
for i=1..len(n) {
```

```
    write(n[i])
```

```
    if ((len(n)-i) mod 3 = 0 and i<n)
```

```
        write(",")
```

```
}
```

Задача С. Число

```
//Pascal
var i,n : integer;
    s : string;
begin
    readln(s);
    n := length(s);
    for i:=1 to n do begin
        write(s[i]);
        if ((n-i) mod 3 = 0) and (i<n) then write(',')
    end
end.
```

Задача D. Снежинка

(Время: 1 сек. Память: 16 Мб Баллы: 100)

Требуется написать программу, печатающую снежинку в квадратной таблице размера $N \times N$, состоящей из символов «.» (точка, ASCII 46) и «*» (звездочка, ASCII 42). Звездочки должны располагаться в средней строке и в среднем столбце, а также на главной и побочной диагоналях.

Входные данные

Входной файл INPUT.TXT содержит натуральное нечетное число N ($N < 1000$) – размер таблицы.

Выходные данные

В выходной файл OUTPUT.TXT выведите снежинку требуемого размера.

Пример

№	INPUT.TXT	OUTPUT.TXT
1	9	<pre>* . . . * . . . * . * . . * . . * . . . * . * . * * * * . . . * * * * * * * * * . . . * * * * . * . * . . . * . . * . . * . * . . . * . . . *</pre>

Система оценки

Решения, работающие только для $N < 20$, будут оцениваться в 30 баллов.

Задача D. Снежинка

Вывод снежинки напоминает вывод квадрата $N \times N$: во вложенном цикле будем для каждой y -й строки выводить символы "." или "*" для каждой x -й позиции. Так, при выводе каждого символа мы будем знать его координату (x, y) .

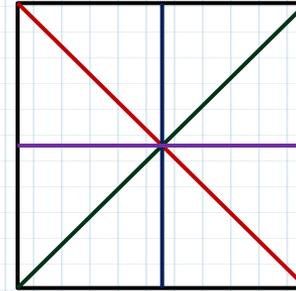
Осталось рассмотреть все случаи, когда следует выводить символ "*":

$x = y$ - главная диагональ

$x = n - y + 1$ - побочная диагональ

$x = n \operatorname{div} 2 + 1$ - центральная вертикаль

$y = n \operatorname{div} 2 + 1$ - центральная горизонталь



Таким образом получаем следующее решение:

```
read(n)
for y=1..n{
  for x=1..n
    if(x=y or x=n-y+1 or y=n div 2+1 or x=n div 2+1)
      write('*')
    else
      write('.')
  writeln
}
```

Задача D. Снежинка

```
//Pascal
var n,i,j : integer;

begin
  read(n);

  for i:=1 to n do begin
    for j:=1 to n do
      if (i=j) or (j=n-i+1) or
         (i=n div 2+1) or (j=n div 2+1)
         then write('*')
         else write('.');

    writeln
  end
end.
```

Задача Е. Правильная таблица

(Время: 1 сек. Память: 16 Мб Баллы: 100)

Будем называть таблицу, состоящую из N строк и M столбцов правильной, если она заполнена целыми числами от 1 до $N \cdot M$ таким образом, что сумма чисел в каждой строке одинакова. При этом числа в таблице не должны повторяться, и каждое из чисел должно присутствовать.

Требуется написать программу, которая по заданным значениям N и M строит правильную таблицу.

Входные данные

Входной файл INPUT.TXT содержит целые числа N и M ($1 \leq N, M \leq 100$) – высоту и ширину таблицы соответственно.

Выходные данные

В выходной файл OUTPUT.TXT выведите N строк по M чисел через пробел в каждой строке – правильную таблицу. Если существует несколько решений, выведите любое. Если решений нет, то следует вывести «No solution» без кавычек.

Примеры

№	INPUT.TXT	OUTPUT.TXT
1	2 4	1 4 6 7 5 3 8 2
2	3 2	1 6 3 4 5 2

Система оценки

Решения, работающие только для четных M , будут оцениваться в 40 баллов.

Задача Е. Правильная таблица

Это самая сложная задача олимпиады, которую далеко не каждый опытный олимпиадник может решить. Однако, 40 баллов набрать относительно не сложно, рассмотрев случай для четного M :

Когда количество столбцов четно, то все столбцы можно разбить на пары так, что в столбцах с нечетными номерами все числа будут идти последовательно по возрастанию, а в четных - по убыванию. Более формально: при нечетном номере столбца j в i -й строке можно выводить число $N \cdot (j-1) + i$, а при четном - $N \cdot (j-1) + n - i + 1$.

Случай для нечетных N и M значительно сложнее. Сразу отметим, что для $M=1$ и $N>1$ в этом случае решения нет. Здесь следует заполнить первые 3 столбца первыми $3 \cdot N$ числами так, чтобы суммы в строках совпадали, а далее можно использовать тот же принцип, что и при четном M , т.к. число оставшихся столбцов четно. В первом столбце поставим в i -й строке число i . Первые $n \div 2$ строк второй строки заполним числами от $2N - N \div 2 + 1$ до $2N$, а оставшиеся - от $N+1$ до $2N - N \div 2$. В третьем столбце числа будут идти по убыванию с шагом 2: сначала первые $N \div 2$ строк заполняются числами от $3N-1$, а со строки $N \div 2 + 1$ идут числа, начиная от $3N$. Так, в каждой строке в первых трех столбцах получаются равные суммы, которые равны $(1+3N) \cdot 3/2$.

Покажем, что при четном N и нечетном M решения не существует. Используя формулу суммы арифметической прогрессии мы можем вычислить сумму, которая должна получаться в каждой строке:

$$S = \frac{1}{N} \cdot \left(\frac{1 + M \cdot N}{2} \cdot M \cdot N \right) = \frac{M + M^2 \cdot N}{2}$$

Сумма чисел в каждой строке - нецелое число, а значит, таблица не может быть правильной.

Случай №1:

$N=6, M=8$ (M - четно)

1	20	21	40	41	60	61	80
2	19	22	39	42	59	62	79
3	18	23	38	43	58	63	78
4	17	24	37	44	57	64	77
5	16	25	36	45	56	65	76
6	15	26	35	46	55	66	75
7	14	27	34	47	54	67	74
8	13	28	33	48	53	68	73
9	12	29	32	49	52	69	72
10	11	30	31	50	51	70	71

Случай №2:

$N=7, M=9$ (N и M - нечетны)

1	12	20	28	29	42	43	56	57
2	13	18	27	30	41	44	55	58
3	14	16	26	31	40	45	54	59
4	8	21	25	32	39	46	53	60
5	9	19	24	33	38	47	52	61
6	10	17	23	34	37	48	51	62
7	11	15	22	35	36	49	50	63

Задача Е. Правильная таблица

```
//Pascal
var n,m,i,j,nj,n2 : integer;
begin
  read(n,m);
  if ((m=1) or odd(m) and not odd(n)) and (n>1) then
    begin write('No solution'); exit end;
  for i:=1 to n do begin
    for j:=1 to m do begin
      nj := (j-1)*n;
      n2 := n div 2;
      if (j=2) and odd(m) then begin
        if i > n2 then write(nj+i-n2, ' ')
          else write(nj+i+n2+1, ' ')
        end else
        if (j=3) and odd(m) then begin
          if i > n2 then write(n*j-2*(i-n2-1), ' ')
            else write(n*j+1-2*i, ' ')
          end else
          if odd(j) then write(nj+i, ' ')
            else write(nj+n-i+1, ' ')
        end;
      writeln
    end
  end
end.
```

Случай №1:

N=6, M=8 (M - четно)

1	20	21	40	41	60	61	80
2	19	22	39	42	59	62	79
3	18	23	38	43	58	63	78
4	17	24	37	44	57	64	77
5	16	25	36	45	56	65	76
6	15	26	35	46	55	66	75
7	14	27	34	47	54	67	74
8	13	28	33	48	53	68	73
9	12	29	32	49	52	69	72
10	11	30	31	50	51	70	71

Случай №2:

N=7, M=9 (N и M - нечетны)

1	12	20	28	29	42	43	56	57
2	13	18	27	30	41	44	55	58
3	14	16	26	31	40	45	54	59
4	8	21	25	32	39	46	53	60
5	9	19	24	33	38	47	52	61
6	10	17	23	34	37	48	51	62
7	11	15	22	35	36	49	50	63