

## Задача А. Фибоначчи

Имя входного файла:

Имя выходного файла:

Ограничение по времени: 1 секунда

Ограничение по памяти: 64 мегабайта

Последовательностью Фибоначчи называется последовательность чисел  $a_0, a_1, \dots, a_n, \dots$ , где  $a_0 = 0, a_1 = 1, a_k = a_{k-1} + a_{k-2}$  при  $(k > 1)$ .

Требуется найти  $N$ -е число Фибоначчи.

В программе запрещается использовать циклы. Программа должна выглядеть примерно так:

**Begin**

```
read(N);
```

```
writeln(fib(N));
```

```
readln;
```

```
readln;
```

**End.**

### Формат входного файла

Водится натуральное число  $N$  ( $1 \leq N \leq 30$ ).

### Формат выходного файла

Выведите  $N$ -е число Фибоначчи.

**Пример**

7	13

## Задача В. НОД

Имя входного файла:

Имя выходного файла:

Ограничение по времени: 1 секунда

Ограничение по памяти: 64 мегабайта

Вводятся 2 числа, хочется посчитать их НОД.

В программе запрещено пользоваться циклами.

Реализуйте функцию `gcd()`, чтобы тело программы имло вид:

**Begin**

```
read(N, M);
```

```
writeln(gcd(N, M));
```

**End.**

### Формат входного файла

Водятся два натуральных числа  $N$  и  $M$ .

### Формат выходного файла

Выведите одно число — НОД( $N, M$ )

**Пример**

9 12	3

## Задача С. Спиралька

Имя входного файла:

Имя выходного файла:

Ограничение по времени: 1 секунда

Ограничение по памяти: 64 мегабайта

Заполните рекурсивно массив числами по спирали, как показано в примерах.

Для этого создайте 4 функции: `fill_right fill_left fill_down fill_up`.

Каждая функция должна принимать 3 параметра: числа  $x, y$  — координату точки, из которой начать заполнение, число  $k$  — первое число, которым нужно заполнить. После этого функция должна заполнить соответствующий столбец/строку и, когда заполнение дойдёт до границы поля или уже заполненной области, рекурсивно вызвать следующую функцию, которая будет заполнять дальше. Программа должна иметь примерно такой вид:

**Begin**

```
read(N);
```

```
<...> { Предобработка, еслионаужна }
```

```
fill_right(1, 1, 1); {заполняем массив}
```

```
{выводим заполненныймассив }
```

```
for i:= 1 to N do
```

**Begin**

```
for j:=1 to N do
```

```
write(map[i, j]:4, ' ');
```

```
writeln;
```

**End;**

End.

### Формат входного файла

Водится натуральное число  $N$  ( $1 \leq N \leq 20$ ).

### Формат выходного файла

Выведите  $N^2$  чисел — заполненный по спирали массив.

### Пример

1	1
2	1 2 4 3
3	1 2 3 8 9 4 7 6 5

## Задача D. Разворот

Имя входного файла:

Имя выходного файла:

Ограничение по времени: 1 секунда

Ограничение по памяти: 64 мегабайта

На ввод подаётся  $N$  чисел. Выведите их в обратном порядке.

Реализуйте функцию `reverse(N)`, вызов которой повлечёт за собой чтение  $N$  чисел и вывод их в обратном порядке. В программе запрещено пользоваться массивами.

### Begin

```
read(N);  
reverse(N);  
readln;  
readln;
```

End.

### Формат входного файла

Водится натуральное число  $N$  ( $1 \leq N \leq 20$ ).

### Формат выходного файла

Выведите оригинальный массив в обратном порядке.

### Пример

3	3 2 1
1 2 3	

## Задача E. Генератор

Имя входного файла:

Имя выходного файла:

Ограничение по времени: 1 секунда

Ограничение по памяти: 64 мегабайта

Даны два натуральных числа  $N$  и  $K$ . Требуется вывести все цепочки  $x_1, x_2, \dots, x_N$  такие, что  $x_i$  - натуральное и  $1 \leq x_i \leq K$ .

Реализуйте рекурсивную функцию генератор, которая выводит на экран все требуемые цепочки.

### Формат входного файла

Водятся натуральныу числа  $N$  и  $M$  ( $1 \leq N, M \leq 6$ ).

### Формат выходного файла

Выведите все требуемые цепочки в произвольном порядке – по одной на строке. Никакая цепочка не должна встречаться более одного раза.

### Пример

2 3	1 1 1 2 1 3 2 1 2 2 2 3 3 1 3 2 3 3

## Задача F. Сумма кубов

Имя входного файла:

Имя выходного файла:

Ограничение по времени: 1 секунда

Ограничение по памяти: 64 мегабайта

Известно, что любое натуральное число можно представить в виде суммы не более чем четырех квадратов натуральных чисел. Вася решил придумать аналогичное утверждение для кубов - он хочет узнать, сколько кубов достаточно для представления любого числа. Его первая рабочая гипотеза - восемь.

Выяснилось, что почти все числа, которые Вася смог придумать, представляются в виде суммы не более чем восьми кубов. Однако число 239, например, не допускает такого представления. Теперь Вася хочет найти какие-либо другие такие числа, а также, возможно, какую-либо закономерность в представлениях всех остальных чисел, чтобы выдвинуть гипотезу относительно вида всех чисел, которые не представляются в виде суммы восьми кубов.

Помогите Васе написать программу, которая проверяла бы, возможно ли представить данное натуральное число в виде суммы не более чем восьми кубов натуральных чисел, и если это возможно, то находила бы какое-либо такое представление.

### Формат входного файла

Вводится натуральное число  $N \leq 2 * 10^9$ .

### Формат выходного файла

требуется вывести не более восьми натуральных чисел, кубы которых в сумме дают  $N$ . Если искомого представления не существует, то в выходной файл необходимо вывести слово IMPOSSIBLE.

### Пример

17	2 2 1
239	IMPOSSIBLE

## Задача G. Ханойские башни

Имя входного файла:

Имя выходного файла:

Ограничение по времени: 1 секунда

Ограничение по памяти: 64 мегабайта

Головоломка "Ханойские башни" состоит из трех стержней, пронумерованных

числами 1, 2, 3. На стержень 1 надета пирамидка из  $n$  дисков различного диаметра в порядке возрастания диаметра. Диски можно перекладывать с одного стержня на другой по одному, при этом диск нельзя класть на диск меньшего диаметра. Необходимо переложить всю пирамидку со стержня 1 на стержень 3 за минимальное число перекладываний.

Напишите программу, которая решает головоломку; для данного числа дисков  $n$  печатает последовательность перекладываний в формате  $a b c$ , где  $a$  — номер перекладываемого диска,  $b$  — номер стержня с которого снимается данный диск,  $c$  — номер стержня на который надевается данный диск.

Например, строка 1 2 3 означает перемещение диска номер 1 со стержня 2 на стержень 3. В одной строке печатается одна команда. Диски пронумерованы числами от 1 до  $n$  в порядке возрастания диаметров.

Программа должна вывести минимальный (по количеству произведенных операций) способ перекладывания пирамидки из данного числа дисков.

### Формат входного файла

### Формат выходного файла

### Пример

2	1 1 2 2 1 3 1 2 3

## Задача H. Ремонт в Ханое

Имя входного файла:

Имя выходного файла:

Ограничение по времени: 1 секунда

Ограничение по памяти: 64 мегабайта

Постановлением ЮНЕСКО оригинал Ханойской башни был подвергнут реставрации. В связи с этим во время пользования головоломкой нельзя было перекладывать кольца с первого стержня сразу на третий и наоборот.

Решите головоломку (переложите все кольца с первого стержня на третий) с учетом этих ограничений. Вам не нужно находить минимальное решение, но количество совершенных перемещений не должно быть больше 200000, при условии, что количество дисков не превосходит 10.

Каждое перемещение задается тремя числами: номер кольца, исходный стержень, конечный стержень.

Формат входного файла

Формат выходного файла

Пример

1	1 1 2 1 2 3
2	1 1 2 1 2 3 2 1 2 1 3 2 1 2 1 2 2 3 1 1 2 1 2 3

### Задача I. Циклические башни

Имя входного файла:

Имя выходного файла:

Ограничение по времени: 1 секунда

Ограничение по памяти: 64 мегабайта

На дорогах Ханоя было введено одностороннее круговое движение, поэтому теперь диск со стержня 1 можно перекладывать только на стержень 2, со стержня 2 на 3, а со стержня 3 на 1.

Решите головоломку с учетом этих ограничений. Вам не нужно находить минимальное решение, но количество совершенных перемещений не должно быть больше 200000, при условии, что количество дисков не превосходит 10.

Формат входного файла

Формат выходного файла

Пример

1	1 1 2 1 2 3
2	1 1 2 1 2 3 2 1 2 1 3 1 2 2 3 1 1 2 1 2 3

### Задача J. Несправедливые башни

Имя входного файла:

Имя выходного файла:

Ограничение по времени: 1 секунда

Ограничение по памяти: 64 мегабайта

В Ханое несправедливо запретили класть самый маленький диск (номер 1) на средний колышек (номер 2).

Решите головоломку с учетом этих ограничений. Вам не нужно находить минимальное решение, но количество совершенных перемещений не должно быть больше 200000, при условии, что количество дисков не превосходит 10.

Формат входного файла

Формат выходного файла

Пример

2	1 1 3 2 1 2 1 3 1 2 2 3 1 1 3