

Задача А. Количество делителей

Имя входного файла: `divisors.in`
Имя выходного файла: `divisors.out`
Ограничение по времени: 2 секунды
Ограничение по памяти: 64 мегабайта

Дано целое число n . Требуется найти количество его делителей, не считая 1 и само число n .

Программа должна содержать функцию

- `function CountDivisors(n: longint): longint` (Pascal),
- `int CountDivisors(int n)` (C++),
- `def CountDivisors(n)` (Python),

возвращающую количество делителей числа n . Программа должна считать число n , вызвать функцию `CountDivisors`, напечатать одну из двух строк в зависимости от результата.

Формат входного файла

Программа получает на вход единственное натуральное число n ($2 \leq n \leq 2^{31} - 1$).

Формат выходного файла

Программа должна выводить одно число — искомое количество делителей.

Примеры

<code>divisors.in</code>	<code>divisors.out</code>
3	0
4	1

Note

Программа, не использующая описанную функцию, засчитываться не будет.

Задача В. Переворот

Имя входного файла: `reverse.in`
Имя выходного файла: `reverse.out`
Ограничение по времени: 2 секунды
Ограничение по памяти: 64 мегабайта

Дано натуральное число N и последовательность из N элементов. Требуется вывести эту последовательность в обратном порядке.

Формат входного файла

В первой строке входного файла записано натуральное число N ($N \leq 500$). В следующих N строках идут N целых чисел, по модулю не превосходящих 1000, — элементы последовательности.

Формат выходного файла

В выходной файл выведите заданную последовательность в обратном порядке. Каждое число должно быть записано на новой строке.

Примеры

<code>reverse.in</code>	<code>reverse.out</code>
3	-3
6	8
8	6
-3	

Note

В этой задаче нельзя использовать циклы и массивы. Даже для считывания.

Задача С. Функция printlist

Имя входного файла: `stdin`
Имя выходного файла: `stdout`
Ограничение по времени: 2 секунды
Ограничение по памяти: 64 мегабайта

Напишите функцию `printlist`, которая получает на вход 4 аргумента:

- список `s`
- строку `before`: что печатать в начале списка
- строку `between`: что печатать между элементами списка
- строку `after`: что печатать после списка

и выводит на экран список `s` в указанном формате. Вставьте вашу функцию в начало следующей программы и сдайте ее на проверку:

```
s = input().split()
printlist(s, 'x=', '*', '.')
printlist(s, '', ' ', '')
printlist(s, '', '', '')
printlist(s, '"', '"', '"', '"')
```

Не забывайте, что список может состоять из 0 или 1 элемента!

Формат входного файла

Формат выходного файла

Примеры

stdin	stdout
abcd e fg	x=abcd*e*fg. abcd e fg abcdefg "abcd "e "fg"

Задача D. Путешествие Нильса с дикими полугусями

Имя входного файла: geese.in
Имя выходного файла: geese.out
Ограничение по времени: 2 секунды
Ограничение по памяти: 64 мегабайта

Когда Нильс путешествовал с дикими гусями, его стая пролетала над озерами. Было решено сделать перерыв, и гуси стали садиться на озёра. На первое озеро села половина всей стаи и ещё полгуся. На второе озеро села половина оставшейся стаи и ещё полгуся. И так далее, пока все гуси не расселись на K озёрах. Теперь Нильс хочет определить, сколько гусей было в стае изначально. Помогите ему это сделать!

Формат входного файла

Во входном файле записано единственное число K ($1 \leq K \leq 20$).

Формат выходного файла

В выходной файл выведите изначально количество гусей в стае.

Примеры

geese.in	geese.out
3	7

Note

Решите задачу, не используя циклы!

Задача E. Ханойские башни

Имя входного файла: hanoi.in
Имя выходного файла: hanoi.out
Ограничение по времени: 6 секунды
Ограничение по памяти: 64 мегабайта

Даны три стержня. На первом стержне находятся несколько дисков сверху вниз по возрастанию размера диска. Два другие пустые. Требуется перенести все диски с первого стержня на второй. Переносить диски разрешается только по одному. Не разрешается класть больший диск на меньший. В программе нельзя пользоваться циклами.

Формат входного файла

Вводится одно число n ($1 \leq n \leq 19$) — количество дисков на первом стержне.

Формат выходного файла

Выведите по два числа в строке — номера стержней, откуда и куда переносится диск. Решение должно быть оптимальным по количеству действий.

Примеры

hanoi.in	hanoi.out
3	1 2 1 3 2 3 1 2 3 1 3 2 1 2