

## Задача А. Реверс

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

Поиски минимума перевернули все вверх дном. Теперь вам надо вернуть все на место. Без циклов. Без списков.

### Формат входного файла

В первой строке — число  $N$ . Во второй строке через пробел указаны  $N$  чисел ( $1 \leq N \leq 900$ ), числа по модулю не превышают  $10^5$ .

### Формат выходного файла

Выведите  $N$  чисел — перевернутый массив.

### Примеры

<code>reverse.in</code>	<code>reverse.out</code>
4 92846 96841 33582 25998	25998 33582 96841 92846

## Задача В. Мегарекурсивный МегаНОД

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

Дано *мало* чисел. Найти самое большое число, на которое делятся все *мало* чисел. Запрещается использование списков и циклов.

### Формат входного файла

В единственной строке даны через пробел *мало* чисел ( $1 \leq \text{мало} \leq 500$ ).

### Формат выходного файла

Выведите искомое число.

### Примеры

<code>megagcd.in</code>	<code>megagcd.out</code>
3 2 3 5	1
3 6 9 12	3

## Задача С. Новый map

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

Разработчикам языка python снова нужна ваша помощь! Благодаря программам, написанным школьниками параллели С' в ЛКШ, функция `map` была признана нестабильной и работающей плохо (выдает странный объект, не позволяет узнавать длину функцией `len`, обнуляется от всякого чиха и т.п.). Напоминаем, что `map` — функция, принимающая на вход два параметра — функцию и список и возвращающая список. Требуется, чтобы программа создала новый список и занесла в него все значения первого списка в том же порядке, предварительно применив к каждому из них функцию. Например `map(lambda x: x * x, [1, 2, 3])` должен вернуть список `[1, 4, 9]`.

### Формат входного файла

Во входном файле в первой строке задана лямбда-функция. Во второй строке через пробел перечислены элементы списка — целые числа.

### Формат выходного файла

Выведите элементы списка после применения функции `map` в том формате, в котором выводит списки python.

### Примеры

<code>make-map.in</code>	<code>make-map.out</code>
<code>lambda x: x * x</code> 2 3 5	<code>[4, 9, 25]</code>

### Note

- Стандартную функцию `map` использовать запрещено.
- Не называйте вашу функцию «`map`». Назовите ее по-другому, например, «`my_map`».

- Чтобы превратить строку в лямбда-функцию, используйте функцию `eval`. Например, `eval('lambda x: x * x')` вернет лямбда-функцию, указанную в условии.

## Задача D. Подмножества

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

Представьте себе, что вы преподаватель. У вас есть  $N$  задач и вам надо выбрать из них контекст. Но пока вы не проверите все варианты, вы не поймете, какой — лучший. Чтобы ничего не забыть, вы решили составить список из всех возможных наборов задач. Естественно, в контексте должна быть хотя бы одна задача.

### Формат входного файла

Дано единственное число  $N$  ( $1 \leq N \leq 10$ ).

### Формат выходного файла

Выведите искомые подмножества множества  $1, 2, \dots, N$ . В каждой строке выведите одно подмножество. Первое число в строке — количество задач в варианте, остальные — номера задач в варианте.

### Примеры

<code>subsequences.in</code>	<code>subsequences.out</code>
1	1 1
2	1 1 1 2 2 1 2