

## Задача А. Минимум на стеке

Имя входного файла: `stack.in`  
Имя выходного файла: `stack.out`  
Ограничение по времени: 1.5 секунды  
Ограничение по памяти: 64 мегабайта

Вам требуется реализовать структуру данных, выполняющую следующие операции:

1. Добавить элемент  $x$  в конец структуры.
2. Удалить последний элемент из структуры.
3. Выдать минимальный элемент в структуре.

### Формат входного файла

В первой строке входного файла задано одно целое число  $n$  — количество операций ( $1 \leq n \leq 10^6$ ). В следующих  $n$  строках заданы сами операции. В  $i$ -ой строке число  $t_i$  — тип операции (1, если операция добавления, 2, если операция удаления, 3, если операция минимума). Если задана операция добавления, то через пробел записано целое число  $x$  — элемент, который следует добавить в структуру ( $-10^9 \leq x \leq 10^9$ ). Гарантируется, что перед каждой операцией удаления или нахождения минимума структура не пуста.

### Формат выходного файла

Для каждой операции нахождения минимума выведите одно число — минимальный элемент в структуре. Ответы разделяйте переводом строки.

### Примеры

<code>stack.in</code>	<code>stack.out</code>
8	-3
1 2	2
1 3	2
1 -3	
3	
2	
3	
2	
3	

## Задача В. Топологическая сортировка

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

Дан ориентированный невзвешенный граф. Необходимо его топологически отсортировать.

### Формат входного файла

В первой строке входного файла даны два целых числа  $N$  и  $M$  ( $1 \leq N \leq 100\,000, 0 \leq M \leq 100\,000$ ) — количества вершин и рёбер в графе соответственно. Далее в  $M$  строках перечислены рёбра графа. Каждое ребро задаётся парой чисел — номерами начальной и конечной вершин соответственно.

### Формат выходного файла

Вывести любую топологическую сортировку графа в виде последовательности номеров вершин. Если граф невозможно топологически отсортировать, вывести «-1».

### Примеры

<code>topsort.in</code>	<code>topsort.out</code>
6 6	4 6 3 1 2 5
1 2	
3 2	
4 2	
2 5	
6 5	
4 6	

## Задача С. НОП-2

Имя входного файла: `lcs2.in`  
Имя выходного файла: `lcs2.out`  
Ограничение по времени: 2 секунды  
Ограничение по памяти: 64 мегабайта

Даны две строки. Найдите их наибольшую общую подпоследовательность.

### Формат входного файла

Во входном файле находятся две строки, состоящие только из маленьких букв английского алфавита. Длина каждой из строк не превышает 1000.

### Формат выходного файла

Выведите одну строку — ответ на задачу.

### Примеры

<code>lcs2.in</code>	<code>lcs2.out</code>
abacaba dacabc	acab
sislksh lkshsis	lksh

### Задача D. Расстояние от точки до прямой

Имя входного файла: `distance1.in`  
Имя выходного файла: `distance1.out`  
Ограничение по времени: 2 секунды  
Ограничение по памяти: 64 мегабайта

Найдите расстояние от заданной точки до заданной прямой.

#### Формат входного файла

Пять целых чисел — координаты точки и коэффициенты  $A$ ,  $B$  и  $C$  нормального уравнения прямой.

#### Формат выходного файла

Одно число — расстояние от точки до прямой с точностью не менее  $10^{-6}$ .

#### Примеры

<code>distance1.in</code>	<code>distance1.out</code>
1 1 1 1 -1	0.7071067812

### Задача E. Количество инверсий

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

Напишите программу, которая для заданного массива  $A = \langle a_1, a_2, \dots, a_n \rangle$  находит количество пар  $(i, j)$  таких, что  $i < j$  и  $a_i > a_j$ .

#### Формат входного файла

Первая строка входного файла содержит натуральное число  $n$  ( $1 \leq n \leq 50\,000$ ) — количество элементов массива. Вторая строка содержит  $n$  попарно различных элементов массива  $A$  — целых неотрицательных чисел, не превосходящих  $10^6$ .

#### Формат выходного файла

В выходной файл выведите одно число — ответ на задачу.

#### Примеры

<code>inverse.in</code>	<code>inverse.out</code>
5 6 11 18 28 31	0
8 999994 999989 999982 999972 999969 999961 999954 999950	28