

### Задача А. Хип ли?

Имя входного файла: `isheap.in`  
Имя выходного файла: `isheap.out`  
Ограничение по времени: 1 секунда  
Ограничение по памяти: 64 мегабайта

Структуру данных *Heap* можно реализовать на основе массива.

Для этого должно выполняться *основное свойство Heap'a*, которое заключается в следующем. Для каждого  $1 \leq i \leq n$  выполняются следующие условия:

- Если  $2i \leq n$ , то  $a[i] \leq a[2i]$
- Если  $2i + 1 \leq n$ , то  $a[i] \leq a[2i + 1]$

Дан массив целых чисел. Определите является ли он *Heap'ом*.

#### Формат входного файла

Первая строка входного файла содержит целое число  $n$  ( $1 \leq n \leq 10^5$ ). Вторая строка содержит  $n$  целых чисел по модулю не превосходящих  $2 \cdot 10^9$ .

#### Формат выходного файла

Выведите «YES», если массив является *Heap'ом* и «NO» в противном случае.

#### Примеры

<code>isheap.in</code>	<code>isheap.out</code>
5 1 0 1 2 0	NO
5 1 3 2 5 4	YES

### Задача В. Хипуй!

Имя входного файла: `heap.in`  
Имя выходного файла: `heap.out`  
Ограничение по времени: 1 секунда  
Ограничение по памяти: 64 мегабайта

В этой задаче вам необходимо организовать структуру данных *Heap* для хранения целых чисел, над которой определены следующие операции:

- `Insert(X)` — добавить в *Heap* число  $X$ ;
- `Extract` — достать из *Heap* наибольшее число (удалив его при этом).

#### Формат входного файла

Во входном файле записано количество команд  $N$  ( $1 \leq N \leq 100\,000$ ), потом последовательность из  $N$  команд, каждая в своей строке.

Каждая команда имеет такой формат: „0 <число>“ или „1“, что означает соответственно операцию `Insert(<число>)` и `Extract`. Добавляемые числа находятся в интервале от 1 до  $10^7$  включительно.

Гарантируется, что при выполнении команды `Extract` в структуре находится по крайней мере один элемент.

#### Формат выходного файла

В выходной файл для каждой команды извлечения необходимо вывести число, полученное при выполнении команды `Extract`.

#### Примеры

<code>heap.in</code>	<code>heap.out</code>
7	100
0 100	50
0 10	
1	
0 5	
0 30	
0 50	
1	

### Задача С. Следующая перестановка

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

Найдите следующую перестановку. Лексикографически первая перестановка является следующей для обратной.

#### Формат входного файла

В первой строке входного файла записано число  $N$  ( $1 \leq N \leq 100\,000$ ) — количество элементов в перестановке. Во второй строке записана перестановка из  $N$  чисел.

#### Формат выходного файла

В выходной файл вывести  $N$  чисел — искомую перестановку.

#### Примеры

<code>nextperm.in</code>	<code>nextperm.out</code>
3 3 2 1	1 2 3
2 1 2	2 1

### Задача D. Номер по перестановке

Имя входного файла: `perm.in`  
Имя выходного файла: `perm.out`  
Ограничение по времени: 1 секунда  
Ограничение по памяти: 64 мегабайта

Дана перестановка из  $N$  чисел от 1 до  $N$ . Требуется найти её номер в лексикографическом порядке.

#### Формат входного файла

Во входном файле сначала записано число  $N$  ( $1 \leq N \leq 12$ ). В следующей строке записана сама перестановка —  $N$  чисел, разделённых пробелами.

#### Формат выходного файла

В выходной файл нужно вывести единственное число — номер перестановки в лексикографическом порядке.

#### Примеры

<code>perm.in</code>	<code>perm.out</code>
3	3
2 1 3	