

D*. Пирамидальная сортировка.

Шаблоны

Ограничение времени	1 секунда
Ограничение памяти	256Mb
Ввод	стандартный ввод или input.txt
Вывод	стандартный вывод или output.txt

Реализуйте две шаблонные функции:

- `*MakeHeap(begin, end)*`, которая принимает указатель на начало и конец последовательности (напомним, конец последовательности - элемент следующий за последним) и переупорядочивает элементы так, чтобы последовательность образовала тах-пирамиду.
- `*SortHeap(begin, end)*`, которая принимает указатель на начало и конец тах-пирамиды и сортирует элементы с помощью пирамидальной сортировки.

Пример:

```
int arr[3, 1, 4, 1, 5, 9];
```

```
MakeHeap(arr, arr + 6); // arr == [9, 4, 5, 1, 1, 3]
```

```
SortHeap(arr, arr + 6); // arr == [1, 1, 3, 4, 5, 9]
```

В качестве решения пришлите заголовочный файл с определением шаблонов. **main писать не нужно - обработка ввода и вывода будет производиться автоматически!**


Формат ввода

В первой строке вводится число N — размер массива ($1 \leq N \leq 10^5$), во второй идут N целых чисел, не превосходящих по модулю 10^9 .


Формат вывода

Выведите отсортированный массив, его элементы необходимо разделять пробелами.

Пример

Ввод 

```
6
1 2 4 6 5 3
```

Вывод 

```
1 2 3 4 5 6
```

