

## 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 входных данных

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 выходных данных

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 входных данных

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 операции **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 выходных данных

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. Коммерческий калькулятор

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

Фирма OISAC выпустила новую версию калькулятора. Этот калькулятор берет с пользователя деньги за совершаемые арифметические операции. Стоимость каждой операции в долларах равна 5% от числа, которое является результатом операции.

На этом калькуляторе требуется вычислить сумму  $N$  натуральных чисел (числа известны). Нетрудно заметить, что от того, в каком порядке мы будем складывать эти числа, иногда зависит, в какую сумму денег нам обойдется вычисление суммы чисел (тем самым, оказывается нарушен классический принцип "от перестановки мест слагаемых сумма не меняется":-) ).

Например, пусть нам нужно сложить числа 10, 11, 12 и 13. Тогда если мы сначала сложим 10 и 11 (это обойдется нам в \$1.05), потом результат - с 12 (\$1.65), и затем - с 13 (\$2.3), то всего мы заплатим \$ 5, если же сначала отдельно сложить 10 и 11 (\$1.05), потом - 12 и 13 (\$1.25) и, наконец, сложить между собой два полученных числа (\$2.3), то в итоге мы заплатим лишь \$4.6.

Напишите программу, которая будет определять, за какую минимальную сумму денег можно найти сумму данных  $N$  чисел.

### Формат входных данных

Во входном файле записано число  $N$  ( $2 \leq N \leq 1000$ ). Далее идет  $N$  натуральных чисел, которые нужно сложить, каждое из них не превышает 10000.

### Формат выходных данных

В выходной файл выведите, сколько денег нам потребуется на нахождение суммы этих  $N$  чисел. Результат должен быть выведен с двумя знаками после десятичной точки.

### Примеры

calc.in	calc.out
4 10 11 12 13	4.60
2 1 1	0.10

### Пример

distance.in	distance.out
4 4 1 3 1 2 1 2 3 2 3 4 5 4 1 4	3

## Задача D. Расстояние между вершинами

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

Коль Дейкстру́ писать без кучи,  
То тайм-лимит ты получишь...  
А в совсем крутой задаче  
Юзай кучу Фибоначчи!

---

Спектакль преподавателей ЛКШ.июль-2007

Дан неориентированный взвешенный граф. Требуется найти вес минимального пути между двумя вершинами.

### Формат входных данных

Первая строка входного файла содержит два натуральных числа  $n$  и  $m$  — количества вершин и рёбер графа соответственно ( $1 \leq n \leq 100\,000$ ,  $1 \leq m \leq 200\,000$ ). Вторая строка входного файла содержит натуральные числа  $s$  и  $t$  — номера вершин, длину пути между которыми требуется найти ( $1 \leq s, t \leq n$ ,  $s \neq t$ ).

Следующие  $m$  строк содержат описание рёбер по одному на строке. Ребро номер  $i$  описывается тремя натуральными числами  $b_i$ ,  $e_i$  и  $w_i$  — номерами концов ребра и его вес соответственно ( $1 \leq b_i, e_i \leq n$ ,  $0 \leq w_i \leq 100$ ).

### Формат выходных данных

Первая строка выходного файла должна содержать одно натуральное число — вес минимального пути между вершинами  $s$  и  $t$ , или -1, если такого пути нет.