

## Задача 1. Вычисление экспоненты

Источник:	повышенной сложности*, <b>обязательна на «отлично»</b>
Имя входного файла:	<code>input.txt</code>
Имя выходного файла:	<code>output.txt</code>
Ограничение по времени:	1 секунда
Ограничение по памяти:	разумное

В данной задаче нужно научиться вычислять экспоненту от заданного числа  $X$ , то есть  $e^X$  для  $e$  — натурального логарифма. Использовать функции `exp`, `pow` и подобные из стандартной библиотеки или откуда-то ещё **запрещено**. Лучше вообще `math.h` не подключать.

**Подсказка:** используйте ряд Тейлора.

### Формат входных данных

В первой строке записано одно целое число  $N$  — количество аргументов, для которых нужно вычислить экспоненту ( $1 \leq N \leq 10^4$ ). Далее идёт  $N$  строк, по одному вещественному числу  $X$  в каждой. Каждое число — это число, экспоненту от которого надо вычислить.

Все числа  $X$  лежат в диапазоне  $(-100, 100)$ .

### Формат выходных данных

Выведите  $N$  строк, в каждой строке одно вещественное число, которое равно  $\exp(X) = e^X$  для соответствующего аргумента  $X$  из входного файла.

Следует выводить числа с помощью формата `"%0.15g"`, чтобы число выводилось в экспоненциальном виде с 15 знаками после десятичной точки. Ответ считается верным, если его относительная ошибка не превышает  $10^{-12}$ .

### Пример

input.txt	output.txt
8	1
0.0	2.71828182845905
1.0	0.367879441171442
-1.0	7.38905609893065
2.0	0.135335283236613
-2.0	2.68811714181614e+43
100.0	3.72007597602084e-44
-100.0	1.20849583696666
0.189376476361643	

## Задача 2. Угол в треугольнике

Источник:	повышенной сложности*, <b>обязательна на «отлично»</b>
Имя входного файла:	<code>input.txt</code>
Имя выходного файла:	<code>output.txt</code>
Ограничение по времени:	1 секунда
Ограничение по памяти:	разумное

Требуется найти угол  $\angle BAC$  в заданном треугольнике  $ABC$ .

### Формат входных данных

В первой строке задано целое число  $Q$  — количество тестовых случаев ( $1 \leq Q \leq 3 \cdot 10^5$ ). В каждой из следующих  $Q$  строк описан один случай.

Каждый случай описывается шестью вещественными числами:  $A_x, A_y$  — координаты вершины угла,  $B_x, B_y$  — координаты другой вершины треугольника,  $C_x, C_y$  — координаты третьей вершины.

Все вещественные числа заданы с максимально возможной точностью и по абсолютной величине не превышают  $10^3$ .

Пусть  $M$  — максимум из абсолютных величин всех шести координат, заданных в тестовом случае. Гарантируется, что  $|AB|, |AC| > \frac{1}{20}M$ . То есть стороны треугольника, инцидентные искомому углу, имеют довольно большую длину.

### Формат выходных данных

Нужно вывести  $Q$  строк, в каждой из которых должен быть записан угол при вершине  $A$  в треугольнике  $ABC$ . Все углы нужно выводить с максимально возможной точностью, рекомендуется использовать формат `"%0.20g"`. Углы нужно выводить в градусах, в пределах от 0 до 180 градусов включительно.

Ответ засчитывается, если он отличается от истинного менее чем на  $10^{-11}$ .

### Пример

input.txt	output.txt
8	36.869897645844019962
2 1 2 3 5 5	45
2 1 4 3 2 3	26.565051177077990019
3 1 3 5 2 3	0
0 0 1 0 10 0	174.28940686250035697
0 0 1 0 -10 1	0
7 4 3 3 3 3	0.00057295779511172474814
0 0 1 0 1 1e-5	89.999427042204885652
0 0 1 0 1e-5 1	

## Задача 3. Динамический поиск

Источник:	повышенной сложности*, <b>обязательна на «отлично»</b>
Имя входного файла:	<code>input.txt</code>
Имя выходного файла:	<code>output.txt</code>
Ограничение по времени:	1 секунда*
Ограничение по памяти:	разумное

Имеется множество целых чисел  $X$ , изначально оно пустое. Нужно выполнить  $M$  заданных операций над этим множеством.

Есть три типа операций:

1. **add**  $v$  — добавить число  $v$  в множество  $X$ . Если такого числа ещё не было в множестве, надо его добавить и напечатать слово **added**. Если такое число уже есть в множестве, нужно напечатать слово **dupe** и ничего не делать.
2. **remove**  $v$  — удалить число  $v$  из множества  $X$ . Если такое число есть в множестве, нужно его удалить и напечатать слово **removed**. А если такого числа нет, нужно напечатать слово **miss** и ничего не делать.
3. **lower**  $v$  — найти минимальное число в множестве  $X$ , которое больше или равно заданному  $v$  (т.е. **lower\_bound**). Если такое число в множестве есть, нужно напечатать в файл. А если его нет, то есть если  $v$  больше всех чисел множества  $X$ , то нужно напечатать **###** (три символа решётки, ASCII 35).

**Внимание:** операции нужно выполнять в режиме “online”: считывать операцию из файла разрешается только после того, как все предыдущие операции уже выполнены.

Задачу нужно решать **используя сбалансированное дерево поиска**.

### Формат входных данных

В первой строке содержится целое число  $M$  — количество операций ( $1 \leq M \leq 3 \cdot 10^5$ ). В остальных  $M$  строках записаны операции в порядке их выполнения. Все числа  $v$  в файле целые и по абсолютной величине не превышают  $10^9$ .

### Формат выходных данных

Нужно вывести  $M$  строк, в каждой из которых требуется записать результат выполнения соответствующей операции.

## Пример

input.txt	output.txt
16	added
add 7	added
add 3	added
add 5	dupe
add 5	added
add 10	dupe
add 7	miss
remove 6	removed
remove 5	added
add 5	removed
remove 3	5
lower 2	5
lower 5	added
add 1	1
lower 0	10
lower 10	###
lower 15	