

Очередь

Реализуйте работу очереди. Для каждой операции изъятия элемента выведите ее результат.

На вход программе подаются строки, содержащие команды. Каждая строка содержит одну команду. Команды бывают двух видов:

- `+ x` – добавить в очередь число x ($-10^9 < x < 10^9$).
- `-` – изъять элемент из очереди. Гарантируется, что не происходит извлечения из пустой очереди.

Гарантируется, что размер очереди в процессе выполнения команд не превысит 10^6 элементов.

Входные данные

В первой строке содержится количество команд n ($1 \leq n \leq 10^6$). Каждая последующая строка исходного файла содержит ровно одну команду.

Выходные данные

Выведите числа, которые удаляются из очереди, по одному в каждой строке.

STDIN

```
4
+ 1
+ 10
-
-
```

STDOUT

```
1
10
```

Примечание

Необходимо написать свою очередь на связном списке.

Стек

Реализуйте работу стека. Для каждой операции изъятия элемента выведите ее результат.

На вход программе подаются строки, содержащие команды. Каждая строка содержит одну команду. Команды бывают двух видов:

- `+ x` – добавить в стек число x ($-10^9 < x < 10^9$).
- `-` – изъять элемент из стека. Гарантируется, что не происходит извлечения из пустого стека.

Гарантируется, что размер стека в процессе выполнения команд не превысит 10^6 элементов.

Входные данные

В первой строке входного файла содержится количество команд n ($1 \leq n \leq 10^6$). Каждая последующая строка исходного файла содержит ровно одну команду.

Выходные данные

Выведите числа, которые удаляются из стека, по одному в каждой строке.

STDIN

```
6
+ 1
+ 10
-
+ 2
+ 1234
-
```

STDOUT

```
10
1234
```

Правильная скобочная последовательность

Входные данные содержат несколько строк, каждая из которых содержит последовательность символов (,), [и]. Выясните, является ли она правильной скобочной последовательностью с двумя типами скобок.

Подсказка: используйте стек.

Входные данные

Не более 500 строк, каждая из которых содержит скобочную последовательность длиной не более чем в 10^4 символов.

Выходные данные

Для каждой строки входных данных выведите в выходной файл YES, если соответствующая последовательность является правильной скобочной последовательностью, или NO, если не является.

STDIN

```
()()
([)]
([)]
(([]
)()
```

STDOUT

```
YES
YES
NO
NO
NO
```

Постфиксная запись

В постфиксной записи (или обратной польской записи) операция записывается после двух операндов. Например, сумма двух чисел A и B записывается как $A\ B\ +$. Запись $B\ C\ +\ D\ *$ обозначает привычное нам $(B + C) * D$, а запись $A\ B\ C\ +\ D\ * +$ означает $A + (B + C) * D$. Достоинство постфиксной записи в том, что она не требует скобок и дополнительных соглашений о приоритете операторов для своего чтения.

Дано выражение в обратной польской записи. Определите его значение. Подсказка: используйте стек.

Входные данные

В единственной строке записано выражение в постфиксной записи, содержащее однозначные числа и операции $+$, $-$, $*$. Строка содержит не более 100 чисел и операций.

Выходные данные

Необходимо вывести значение записанного выражения. Гарантируется, что результат выражения, а также результаты всех промежуточных вычислений по модулю меньше 2^{31} .

STDIN

8 9 + 1 7 - *

STDOUT

-102

Макстек

Реализуйте работу Макстека (стека, в котором можно найти текущий максимум). Для каждой операции нахождения максимума выведите ее результат.

На вход программе подаются строки, содержащие команды. Каждая строка содержит одну команду. Команды бывают трёх видов:

- `+` x – добавить в стек число x ($1 < x < 10^9$).
- `-` – изъять элемент из стека. Гарантируется, что не происходит извлечения из пустого стека.
- `max` – найти максимальное число, содержащееся в стеке.

Гарантируется, что размер стека в процессе выполнения команд не превысит 10^6 элементов.

Входные данные

В первой строке входного файла содержится количество команд n ($1 \leq n \leq 10^6$). Каждая последующая строка исходного файла содержит ровно одну команду.

Выходные данные

В ответ на каждую команду `max` выведите максимальное число, содержащееся в стеке на момент получения команды.

STDIN

```
5
+ 2
+ 1
max
-
max
```

STDOUT

```
2
2
```

Бинарный поиск

Дан массив из n элементов, упорядоченный в порядке неубывания, и m запросов: найти первое и последнее вхождение числа в массив.

Входные данные

В первой строке содержится одно число n – размер массива ($1 \leq n \leq 100000$).

Во второй строке находится n чисел в порядке неубывания – элементы массива.

В третьей строке находится число m – количество запросов.

В последней строке находится m чисел – запросы.

Выходные данные

Для каждого запроса выведите в отдельной строке номер первого и последнего вхождения этого числа в массив. Если числа в массиве нет, выведите `-1 -1`.

STDIN

```
5
1 1 2 2 2
3
1 2 3
```

STDOUT

```
1 2
3 5
-1 -1
```

Гирлянда

Гирлянда состоит из n лампочек на общем проводе. Один её конец закреплён на заданной высоте A мм ($h_1 = A$). Благодаря силе тяжести гирлянда прогибается: высота каждой неконцевой лампы на 1 мм меньше, чем средняя высота ближайших соседей ($h_i = (h_{i-1} + h_{i+1}) \div 2 - 1$ для $1 < i < n$).




Требуется найти минимальную высоту второго конца B ($B = h_n$) при условии, что ни одна из лампочек не должна лежать на земле ($h_i > 0$ для $1 \leq i \leq n$).

Входные данные

В единственной строке содержатся числа n и A ($3 \leq n \leq 1000$ - целое, $10 \leq A \leq 1000$ - вещественное).

Выходные данные

Вывести одно вещественное число B с двумя знаками после запятой.

STDIN		STDOUT	
8 15		9.75	
692 532.81		446113.34	