

В данной лабораторной вам предстоит сдавать задачи в PCMS. Можете использовать логины и пароли, которые вы использовали раньше для лабораторных по АиСД. Длительность лабораторной ровно одна неделя. Перевод этой лабораторной в нашу обычную систему счисления будет следующий:

0. Если в задаче вы использовали самописные структуры данных — задача будет аннулирована по окончании лабораторной.
1. Если вы не сдали 3 и более задач — лабораторная не засчитывается.
2. За каждую несданную задачу вы получаете один **Delay**.
3. За каждую неправильную посылку вы получаете один **minus**.
4. Если вы получили **OK**, то за каждую последующую посылку по этой задаче, которая пройдет первый тест, вы получаете **minus**.

Enjoy!

## Задача А. Сбалансированное двоичное дерево поиска

Имя входного файла:            стандартный ввод  
Имя выходного файла:        стандартный вывод  
Ограничение по времени:    0.1 секунды  
Ограничение по памяти:      256 мегабайт

Реализуйте сбалансированное двоичное дерево поиска.

### Формат входного файла

Входной файл содержит описание операций с деревом, их количество не превышает  $10^5$ . В каждой строке находится одна из следующих операций:

- **insert**  $x$  — добавить в дерево ключ  $x$ . Если ключ  $x$  есть в дереве, то ничего делать не надо
- **delete**  $x$  — удалить из дерева ключ  $x$ . Если ключа  $x$  в дереве нет, то ничего делать не надо
- **exists**  $x$  — если ключ  $x$  есть в дереве выведите «true», если нет «false»
- **next**  $x$  — выведите минимальный элемент в дереве, строго больший  $x$ , или «none» если такого нет
- **prev**  $x$  — выведите максимальный элемент в дереве, строго меньший  $x$ , или «none» если такого нет

В дерево помещаются и извлекаются только целые числа, не превышающие по модулю  $10^9$ .

### Формат выходного файла

Выведите последовательно результат выполнения всех операций **exists**, **next**, **prev**. Следуйте формату выходного файла из примера.

### Пример

стандартный ввод	стандартный вывод
insert 2	true
insert 5	false
insert 3	5
exists 2	3
exists 4	none
next 4	3
prev 4	
delete 5	
next 4	
prev 4	

## Задача В. Перестановки

Имя входного файла:            стандартный ввод  
Имя выходного файла:        стандартный вывод  
Ограничение по времени:    0.75 секунды  
Ограничение по памяти:      256 мегабайт

Во входном файле задано число  $n$  ( $1 \leq n \leq 9$ ). Выведите в выходной файл в лексикографическом порядке все перестановки чисел от 1 до  $n$ .

### Пример

стандартный ввод	стандартный вывод
3	1 2 3 1 3 2 2 1 3 2 3 1 3 1 2 3 2 1

## Задача С. Очередь

Имя входного файла:            стандартный ввод  
Имя выходного файла:        стандартный вывод  
Ограничение по времени:    0.4 секунды  
Ограничение по памяти:      256 мегабайт

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

На вход программе подаются строки, содержащие команды. Каждая строка содержит одну команду. Команда — это либо “+ N”, либо “-”. Команда “+ N” означает добавление в очередь числа  $N$ , по модулю не превышающего  $10^9$ . Команда “-” означает изъятие элемента из очереди.

### Формат входного файла

В первой строке содержится количество команд —  $M$  ( $1 \leq M \leq 10^6$ ). В последующих строках содержатся команды, по одной в каждой строке.

### Формат выходного файла

Выведите числа, которые удаляются из очереди, по одному в каждой строке. Гарантируется, что извлечения из пустой очереди не производится.

### Пример

стандартный ввод	стандартный вывод
4	1
+ 1	10
+ 10	
-	
-	

## Задача D. Стек

Имя входного файла:            стандартный ввод  
Имя выходного файла:        стандартный вывод  
Ограничение по времени:    0.4 секунды  
Ограничение по памяти:      256 мегабайт

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

На вход программе подаются строки, содержащие команды. Каждая строка содержит одну команду. Команда — это либо “+ N”, либо “-”. Команда “+ N” означает добавление в стек числа  $N$ , по модулю не превышающего  $10^9$ . Команда “-” означает изъятие элемента из стека. Гарантируется, что не происходит извлечения из пустого стека.

### Формат входного файла

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

### Формат выходного файла

Выведите числа, которые удаляются из стека, по одному в каждой строке. Гарантируется, что изъятий из пустого стека не производится.

### Пример

стандартный ввод	стандартный вывод
6	10
+ 1	1234
+ 10	
-	
+ 2	
+ 1234	
-	

## Задача Е. К-ая порядковая статистика

Имя входного файла:            стандартный ввод  
Имя выходного файла:        стандартный вывод  
Ограничение по времени:    0.4 секунды  
Ограничение по памяти:      256 мегабайт

Дан массив из  $n$  элементов. Какое число  $k$ -ое в порядке возрастания в этом массиве.

### Формат входного файла

В первую строке входного файла содержится два числа  $n$  — размер массива и  $k$ . ( $1 \leq k \leq n \leq 3 \cdot 10^7$ ). Во второй строке находятся числа  $A, B, C, a_1, a_2$  по модулю не превосходящие  $10^9$ . Вы должны получить элементы массива начиная с третьего по формуле:  $a_i = A * a_{i-2} + B * a_{i-1} + C$ . Все вычисления должны производиться в 32 битном знаковом типе, переполнения должны игнорироваться.

### Формат выходного файла

Выведите значение  $k$ -ое в порядке возрастания число в массиве  $a$ .

### Пример

стандартный ввод	стандартный вывод
5 3 2 3 5 1 2	13
5 3 200000 300000 5 1 2	2

Во втором примере элементы массива  $a$  равны: (1, 2, 800005, −516268571, 1331571109).

## Задача F. Двоичный поиск

Имя входного файла:            стандартный ввод  
Имя выходного файла:        стандартный вывод  
Ограничение по времени:    0.62 секунды  
Ограничение по памяти:      256 мегабайт

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

### Формат входного файла

В первую строке входного файла содержится одно число  $n$  — размер массива ( $1 \leq n \leq 500000$ ). Во второй строке находится  $n$  чисел в порядке неубывания — элементы массива. В третьей строке находится число  $m$  — количество запросов ( $1 \leq m \leq 500000$ ). В следующей строке находится  $m$  чисел — запросы.

### Формат выходного файла

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

### Пример

стандартный ввод	стандартный вывод
5	1 2
1 1 2 2 2	3 5
3	-1 -1
1 2 3	

## Задача G. Тестирующая система

Имя входного файла:            стандартный ввод  
Имя выходного файла:        стандартный вывод  
Ограничение по времени:    0.1 секунды  
Ограничение по памяти:      256 мегабайт

Юный программист Саша написал свою первую тестирующую систему. Он так обрадовался тому, что она скомпилировалась, что решил пригласить школьных друзей на свой собственный констест.

Но в конце тура выяснилось, что система не умеет сортировать команды в таблице результатов. Помогите Саше реализовать эту сортировку.

Команды упорядочиваются по правилам ACM:

- по количеству решённых задач в порядке убывания;
- при равенстве количества решённых задач — по штрафному времени в порядке возрастания;
- при прочих равных — по номеру команды в порядке возрастания.

### Формат входного файла

Первая строка содержит натуральное число  $n$  ( $1 \leq n \leq 100\,000$ ) — количество команд, участвующих в констесте. В  $i$ -й из следующих  $n$  строк записано количество решенных задач  $S$  ( $1 \leq S \leq 100$ ) и штрафное время  $T$  ( $1 \leq T \leq 100\,000$ ) команды с номером  $i$ .

### Формат выходного файла

В выходной файл выведите  $n$  чисел — номера команд в отсортированном порядке.

### Пример

стандартный ввод	стандартный вывод
5 3 50 5 720 1 7 0 0 8 500	5 2 1 3 4



## Задача H. Set

Имя входного файла:	стандартный ввод
Имя выходного файла:	стандартный вывод
Ограничение по времени:	0.58 секунды
Ограничение по памяти:	256 мегабайт

Реализуйте множество.

### Формат входного файла

Входной файл содержит описание операций, их количество не превышает 1000000. В каждой строке находится одна из следующих операций:

- `insert x` — добавить элемент  $x$  в множество. Если элемент уже есть в множестве, то ничего делать не надо.
- `delete x` — удалить элемент  $x$ . Если элемента  $x$  нет, то ничего делать не надо.
- `exists x` — если ключ  $x$  есть в множестве выведите «true», если нет «false».

В множество помещаются и извлекаются только целые числа, не превышающие по модулю  $10^9$ .

### Формат выходного файла

Выведите последовательно результат выполнения всех операций `exists`. Следуйте формату выходного файла из примера.

### Пример

стандартный ввод	стандартный вывод
<code>insert 2</code>	<code>true</code>
<code>insert 5</code>	<code>false</code>
<code>insert 3</code>	<code>false</code>
<code>exists 2</code>	
<code>exists 4</code>	
<code>insert 2</code>	
<code>delete 2</code>	
<code>exists 2</code>	

## Задача I. Мар

Имя входного файла:	стандартный ввод
Имя выходного файла:	стандартный вывод
Ограничение по времени:	1.15 секунды
Ограничение по памяти:	256 мегабайт

Реализуйте ассоциативный массив.

### Формат входного файла

Входной файл содержит описание операций, их количество не превышает 1000000. В каждой строке находится одна из следующих операций:

- `put x y` — поставить в соответствие ключу `x` значение `y`. Если ключ уже есть, то значение необходимо изменить.
- `delete x` — удалить ключ `x`. Если элемента `x` нет, то ничего делать не надо.
- `get x` — если ключ `x` есть в ассоциативном массиве, то выведите соответствующее ему значение, иначе выведите «none».

Ключи и значения — строки из латинских букв длиной не более 20 символов.

### Формат выходного файла

Выведите последовательно результат выполнения всех операций `get`. Следуйте формату выходного файла из примера.

### Пример

стандартный ввод	стандартный вывод
put hello privet put bye пока get hello get bye delete hello get hello	privet пока none

## Задача J. $K$ -й максимум

Имя входного файла:            стандартный ввод  
Имя выходного файла:        стандартный вывод  
Ограничение по времени:    0.15 секунды  
Ограничение по памяти:      256 мегабайт

Напишите программу, реализующую структуру данных, позволяющую добавлять и удалять элементы, а также находить  $k$ -й максимум.

### Формат входного файла

Первая строка входного файла содержит натуральное число  $n$  — количество команд ( $n \leq 100\,000$ ). Последующие  $n$  строк содержат по одной команде каждая. Команда записывается в виде двух чисел  $c_i$  и  $k_i$  — тип и аргумент команды соответственно ( $|k_i| \leq 10^9$ ). Поддерживаемые команды:

- $+1$  (или просто  $1$ ): Добавить элемент с ключом  $k_i$ .
- $0$ : Найти и вывести  $k_i$ -й максимум.
- $-1$ : Удалить элемент с ключом  $k_i$ .

Гарантируется, что в процессе работы в структуре не требуется хранить элементы с равными ключами или удалять несуществующие элементы. Также гарантируется, что при запросе  $k_i$ -го максимума, он существует.

### Формат выходного файла

Для каждой команды нулевого типа в выходной файл должна быть выведена строка, содержащая единственное число —  $k_i$ -й максимум.

### Пример

стандартный ввод	стандартный вывод
11	7
+1 5	5
+1 3	3
+1 7	10
0 1	7
0 2	3
0 3	
-1 5	
+1 10	
0 1	
0 2	
0 3	

## Задача К. Переместить в начало

Имя входного файла:            стандартный ввод  
Имя выходного файла:        стандартный вывод  
Ограничение по времени:    2.2 секунды  
Ограничение по памяти:      256 мегабайт

Вам дан массив  $a_1 = 1, a_2 = 2, \dots, a_n = n$  и последовательность операций: переместить элементы с  $l_i$  по  $r_i$  в начало массива. Например, для массива 2, 3, 6, 1, 5, 4, после операции (2, 4) новый порядок будет 3, 6, 1, 2, 5, 4. А после применения операции (3, 4) порядок элементов в массиве будет 1, 2, 3, 6, 5, 4.

Выведите порядок элементов в массиве после выполнения всех операций.

### Формат входного файла

В первой строке входного файла указаны числа  $n$  и  $m$  ( $2 \leq n \leq 100\,000$ ,  $1 \leq m \leq 100\,000$ ) — число элементов в массиве и число операций. Следующие  $m$  строк содержат операции в виде двух целых чисел:  $l_i$  и  $r_i$  ( $1 \leq l_i \leq r_i \leq n$ ).

### Формат выходного файла

Выведите  $n$  целых чисел — порядок элементов в массиве после применения всех операций.

### Пример

стандартный ввод	стандартный вывод
6 3 2 4 3 5 2 2	1 4 5 2 3 6