

## Задача А. Трехвидовая ПСП

Имя входного файла:            стандартный ввод  
Имя выходного файла:        стандартный вывод  
Ограничение по времени:    1 секунда  
Ограничение по памяти:      256 мегабайт

На лекции вы обсуждали понятие правильности для скобочной последовательности с более чем одним видом скобок. Ваша задача — проверить скобочную последовательность, если она состоит из следующих видов скобок:  $(, ), [, ], \{, \}$ .

### Формат входных данных

Дана скобочная последовательность из символов выше длиной не более  $10^5$ .

### Формат выходных данных

Выведите YES, если скобочная последовательность правильная, и NO — иначе.

### Примеры

стандартный ввод	стандартный вывод
<code>() []</code>	YES
<code>([])</code>	NO

## Задача В. Min queue

Имя входного файла:	стандартный ввод
Имя выходного файла:	стандартный вывод
Ограничение по времени:	2 секунды
Ограничение по памяти:	256 мегабайт

Реализуйте структуру данных «очередь» с помощью двух динамических структур данных «стек», реализованных ранее.

Напишите программу, содержащую описание очереди и моделирующую работу очереди, реализовав все указанные здесь методы. Возможные команды для программы:

- «enqueue  $n$ » Добавить в хвост очереди число  $n$  ( $1 \leq n \leq 10^9$ ) (значение  $n$  задается после команды). Программа должна вывести «ok».
- «dequeue» Удалить из очереди головной элемент. Программа должна вывести его значение.
- «front» Программа должна вывести значение головного элемента, не удаляя его из очереди.
- «size» Программа должна вывести количество элементов в очереди.
- «clear» Программа должна очистить очередь и вывести «ok».
- «min» Программа должна вывести минимум в очереди, не удаляя его из очереди.

Перед исполнением операций «front», «dequeue» и «min» программа должна проверять, содержится ли в очереди хотя бы один элемент. Если во входных данных встречается операция «front», «dequeue» или «min», и при этом очередь пуста, то программа должна вместо числового значения вывести строку «error».

### Формат входных данных

В первой строке входных данных записано единственное число  $M$  ( $1 \leq M \leq 2 \cdot 10^5$ ) — количество команд. В следующих строках дано по одной команде из тех, что идут выше.

### Формат выходных данных

Для каждой команды выведите одну строку — результат ее выполнения.

### Пример

стандартный ввод	стандартный вывод
9	ok
enqueue 2	2
front	2
dequeue	0
size	error
dequeue	ok
enqueue 1	ok
enqueue 2	2
size	1
min	

## Задача С. Дек

Имя входного файла: стандартный ввод  
Имя выходного файла: стандартный вывод  
Ограничение по времени: 1 секунда  
Ограничение по памяти: 256 мегабайт

Реализуйте динамическую структуру данных «дек».

Программа считывает последовательность команд и в зависимости от команды выполняет ту или иную операцию. После выполнения каждой команды программа должна вывести одну строчку. Возможные команды для программы:

- *push\_front* — Добавить (положить) в начало дека новый элемент. Программа должна вывести *ok*.
- *push\_back* — Добавить (положить) в конец дека новый элемент. Программа должна вывести *ok*.
- *pop\_front* — Извлечь из дека первый элемент. Программа должна вывести его значение.
- *pop\_back* — Извлечь из дека последний элемент. Программа должна вывести его значение.
- *front* — Узнать значение первого элемента (не удаляя его). Программа должна вывести его значение.
- *back* — Узнать значение последнего элемента (не удаляя его). Программа должна вывести его значение.
- *size* — Вывести количество элементов в деке.
- *clear* — Очистить дек (удалить из него все элементы) и вывести *ok*.
- *exit* — Программа должна закончить свою работу и вывести *bye*.

Перед исполнением операций *pop\_front*, *pop\_back*, *front*, *back* программа должна проверять, содержится ли в деке хотя бы один элемент. Если во входных данных встречается операция *pop\_front*, *pop\_back*, *front*, *back*, и при этом дек пуст, то программа должна вместо числового значения вывести строку *error*.

### Формат входных данных

Вводятся команды управления деком, по одной на строке.

### Формат выходных данных

Требуется вывести протокол работы дека, по одному сообщению на строке.

### Примеры

стандартный ввод	стандартный вывод
push_back 1 back exit	ok 1 bye
size push_back 1 size push_back 2 size push_front 3 size exit	0 ok 1 ok 2 ok 3 bye

## Задача D. Польская запись

Имя входного файла: стандартный ввод  
Имя выходного файла: стандартный вывод  
Ограничение по времени: 1 секунда  
Ограничение по памяти: 256 мегабайт

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.

### Формат входных данных

В единственной строке записано выражение в постфиксной записи, содержащее однозначные числа и операции « $+$ », « $-$ », « $*$ ».

### Формат выходных данных

Необходимо вывести значение записанного выражения.

### Пример

стандартный ввод	стандартный вывод
8 9 + 1 7 - *	-102

## Задача Е. Гоблины

Имя входного файла:	стандартный ввод
Имя выходного файла:	стандартный вывод
Ограничение по времени:	1 секунда
Ограничение по памяти:	256 мегабайт

Гоблины Мглистых гор очень любят ходить к своим шаманам. Так как гоблинов много, к шаманам часто образуются очень длинные очереди. А поскольку много гоблинов в одном месте быстро образуют шумную толку, которая мешает шаманам проводить сложные медицинские манипуляции, последние решили установить некоторые правила касательно порядка в очереди.

Обычные гоблины при посещении шаманов должны вставать в конец очереди. Привилегированные же гоблины, знающие особый пароль, встают ровно в ее середину, причем при нечетной длине очереди они встают сразу за центром.

Так как гоблины также широко известны своим непочтительным отношением ко всяческим правилам и законам, шаманы попросили вас написать программу, которая бы отслеживала порядок гоблинов в очереди.

### Формат входных данных

В первой строке входных данных записано число  $N$  ( $1 \leq N \leq 10^5$ ) — количество запросов к программе. Следующие  $N$  строк содержат описание запросов в формате:

- «+  $i$ » — гoblin с номером  $i$  ( $1 \leq i \leq N$ ) встает в конец очереди;
- «\*  $i$ » — привилегированный гoblin с номером  $i$  встает в середину очереди;
- «-» — первый гoblin из очереди уходит к шаманам. Гарантируется, что на момент такого запроса очередь не пуста.

### Формат выходных данных

Для каждого запроса типа «-» программа должна вывести номер гоблина, который должен уйти к шаманам.

### Примеры

стандартный ввод	стандартный вывод
7 + 1 + 2 - + 3 + 4 - -	1 2 3
10 + 1 + 2 * 3 - + 4 * 5 - - - -	1 3 2 5 4

## Задача F. Карточная игра

Имя входного файла:	стандартный ввод
Имя выходного файла:	стандартный вывод
Ограничение по времени:	1 секунда
Ограничение по памяти:	256 мегабайт

В игре в пьяницу карточная колода раздается поровну двум игрокам. Далее они вскрывают по одной верхней карте, и тот, чья карта старше, забирает себе обе вскрытые карты, которые кладутся под низ его колоды. Тот, кто остается без карт — проигрывает.

Для простоты будем считать, что все карты различны по номиналу, а также, что самая младшая карта побеждает самую старшую карту («шестерка берет туза»).

Игрок, который забирает себе карты, сначала кладет под низ своей колоды карту первого игрока, затем карту второго игрока (то есть карта второго игрока оказывается внизу колоды).

Напишите программу, которая моделирует игру в пьяницу и определяет, кто выигрывает. В игре участвует 10 карт, имеющих значения от 0 до 9, большая карта побеждает меньшую, карта со значением 0 побеждает карту 9.

### Формат входных данных

Программа получает на вход две строки: первая строка содержит 5 чисел, разделенных пробелами — номера карт первого игрока, вторая — аналогично 5 карт второго игрока. Карты перечислены сверху вниз, то есть каждая строка начинается с той карты, которая будет открыта первой.

### Формат выходных данных

Программа должна определить, кто выигрывает при данной раздаче, и вывести слово «first» или «second», после чего вывести количество ходов, сделанных до выигрыша. Если на протяжении  $10^6$  ходов игра не заканчивается, программа должна вывести слово «botva».

### Пример

стандартный ввод	стандартный вывод
1 3 5 7 9 2 4 6 8 0	second 5

## Задача Г. Быстрая сортировка

Имя входного файла: стандартный ввод  
Имя выходного файла: стандартный вывод  
Ограничение по времени: 1 секунда  
Ограничение по памяти: 256 мегабайт

Нужно с помощью быстрой сортировки отсортировать массив.

### Формат входных данных

В первой строке дано число  $N$  ( $1 \leq N \leq 10^5$ ). Далее во второй строке идет  $N$  чисел через пробел, числа не превосходят по модулю  $10^9$ .

### Формат выходных данных

Вывести отсортированный по неубыванию массив.

### Пример

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

## Задача Н. $k$ -я порядковая статистика

Имя входного файла: стандартный ввод  
Имя выходного файла: стандартный вывод  
Ограничение по времени: 1 секунда  
Ограничение по памяти: 256 мегабайт

В этой задаче вам необходимо найти  $k$ -ую порядковую статистику ( $k$ -ое по неубыванию) числовой последовательности  $A$ , элементы которой задаются следующим образом:

$$A_i = (A_{i-1} \cdot 123 + A_{i-1} \cdot 45) \% (10^7 + 4321)$$

### Формат входных данных

Вам даны  $n, k, A_0, A_1$  ( $1 \leq k \leq n \leq 10^7, 0 \leq A_i < 10^7 + 4321$ ) — количество чисел в числовой последовательности,  $k$  из условия и первые два числа числовой последовательности.

### Формат выходных данных

Выведите  $k$ -ую порядковую статистику.

### Примеры

стандартный ввод	стандартный вывод
2 2 100 1000	1000
10 5 1 2	321287



## Задача I. Приоритетная очередь

Имя входного файла: стандартный ввод  
Имя выходного файла: стандартный вывод  
Ограничение по времени: 1 секунда  
Ограничение по памяти: 256 мегабайт

Задача 4 отсюда [https://informatics.msk.ru/file.php/18/bin\\_heap\\_ig-091022.pdf](https://informatics.msk.ru/file.php/18/bin_heap_ig-091022.pdf)

### Пример

стандартный ввод	стандартный вывод
4 7	-1
1	1
2 9	2
2 4	3
2 9	2
2 9	-1
2 7	2 9
1	9 4 9

## Задача J. Приоритетная очередь с удалением

Имя входного файла: стандартный ввод  
Имя выходного файла: стандартный вывод  
Ограничение по времени: 1 секунда  
Ограничение по памяти: 256 мегабайт

Задача 5 в документе по ссылке: [https://informatics.msk.ru/file.php/18/bin\\_heap\\_ig-091022.pdf](https://informatics.msk.ru/file.php/18/bin_heap_ig-091022.pdf)

### Пример

стандартный ввод	стандартный вывод
4 10	-1
1	1
2 9	2
2 4	3
2 9	2
2 9	-1
2 7	2 9
1	-1
3 4	4
2 1	9
3 3	9 4 1

## Задача К. Пирамидальная сортировка

Имя входного файла: стандартный ввод  
Имя выходного файла: стандартный вывод  
Ограничение по времени: 1 секунда  
Ограничение по памяти: 256 мегабайт

Требуется отсортировать массив с помощью алгоритма пирамидальной сортировки. Решение необходимо оформить в виде функции `void HeapSort(int* begin, int* end)`, где `begin`, `end` — указатели на начало и конец массива, который надо отсортировать. В конце работы программа должна вывести отсортированный массив.

### Формат входных данных

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 выходных данных

Выведите отсортированный массив, его элементы необходимо разделять пробелами.

### Пример

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

## Задача L. Пирамида

Имя входного файла:	стандартный ввод
Имя выходного файла:	стандартный вывод
Ограничение по времени:	2 секунды
Ограничение по памяти:	256 мегабайт

Реализуйте структуру данных «пирамида». В пирамиде лучше всего хранить не сами значения, а номера значений в порядке добавления. Напишите программу, реализовав все указанные здесь методы. Возможные команды для программы:

- «insert  $n$ » — Добавить в пирамиду число  $n$  ( $1 \leq n \leq 10^9$ ) (значение  $n$  задается после команды). Программа должна вывести **ok**;
- «extract\_min» — Удалить из пирамиды минимальный элемент. Программа должна вывести его значение;
- «delete  $x$ » Удалить из пирамиды элемент, который был добавлен  $x$ -м по счёту командой insert ( $1 \leq x \leq 200000$ ). Программа должна вывести **ok**;
- «change  $x$   $n$ » — Изменить значение элемента, который был добавлен  $x$ -м по счёту командой insert ( $1 \leq x \leq 200000$ ) на  $n$  ( $1 \leq n \leq 10^9$ ). Программа должна вывести **ok**.
- «get\_min» — Программа должна вывести значение минимального элемента, не удаляя его из пирамиды;
- «size» — Программа должна вывести количество элементов в пирамиде;
- «clear» — Программа должна очистить пирамиду и вывести **ok**.

Гарантируется, что в каждый момент времени в пирамиде содержатся только различные элементы.

Перед исполнением операций **extract\_min** и **get\_min** программа должна проверять, содержится ли в пирамиде хотя бы один элемент. Если во входных данных встречается операция **extract\_min** или **get\_min**, и при этом пирамида пуста, то программа должна вместо числового значения вывести строку **error**.

Перед исполнением операций **delete** и **change** программа должна проверять, содержится ли в пирамиде  $x$ -й элемент. Если во входных данных встречается операция **delete** или **change**, и при этом соответствующего элемента  $x$  в пирамиде нет, то программа должна вместо **ok** вывести строку **error**.

### Формат входных данных

В первой строке входных данных записано единственное число  $M$  ( $1 \leq M \leq 200000$ ) — количество команд. В следующих строках дано по одной команде из тех, что идут выше.

### Формат выходных данных

Для каждой команды выведите одну строку — результат ее выполнения.

## Пример

стандартный ввод	стандартный вывод
20	ok
insert 18	18
extract_min	ok
insert 3	3
extract_min	ok
insert 10	10
extract_min	ok
insert 14	ok
insert 20	ok
insert 11	ok
delete 6	14
extract_min	ok
insert 8	8
get_min	ok
insert 9	ok
change 7 12	error
change 4 5	ok
insert 13	ok
change 5 16	ok
insert 19	ok
insert 7	

## Задача М. Минимакс

Имя входного файла:	стандартный ввод
Имя выходного файла:	стандартный вывод
Ограничение по времени:	2 секунды
Ограничение по памяти:	256 мегабайт

Реализуйте структуру данных «две пирамиды». Напишите программу, реализовав все указанные здесь методы. Возможные команды для программы:

- *insert n* — добавить в структуру число  $n$  ( $1 \leq n \leq 10^9$ ) (значение  $n$  задается после команды). Программа должна вывести **ok**.
- *extract\_min* — удалить из структуры минимальный элемент. Программа должна вывести его значение.
- *get\_min* — программа должна вывести значение минимального элемента, не удаляя его из структуры.
- *extract\_max* — удалить из структуры максимальный элемент. Программа должна вывести его значение.
- *get\_max* — программа должна вывести значение максимального элемента, не удаляя его из структуры.
- *size* — программа должна вывести количество элементов в структуре.
- *clear* — Программа должна очистить структуру и вывести **ok**.

Перед исполнением операций *extract\_min*, *extract\_max*, *get\_min* и *get\_max* программа должна проверять, содержится ли в структуре хотя бы один элемент.

Если во входных данных встречается операция *extract\_min*, *extract\_max*, *get\_min* или *get\_max*, и при этом в структуре нет ни одного элемента, то программа должна вместо числового значения вывести строку **error**.

### Формат входных данных

В первой строке входных данных записано единственное число  $M$  ( $1 \leq M \leq 2 \cdot 10^5$ ) — количество команд.

В следующих строках дано по одной команде из тех, что идут выше.

### Формат выходных данных

Для каждой команды выведите одну строчку — результат ее выполнения.

## Пример

стандартный ввод	стандартный вывод
16	ok
insert 7	7
extract_min	ok
insert 1	ok
insert 2	2
extract_max	ok
insert 2	ok
insert 6	3
size	ok
insert 9	ok
insert 7	9
get_max	5
size	ok
clear	error
get_max	ok
insert 8	8
extract_max	

## Задача N. MinMidMax

Имя входного файла:            стандартный ввод  
Имя выходного файла:        стандартный вывод  
Ограничение по времени:    1 секунда  
Ограничение по памяти:      256 мегабайт

Реализуйте структуру данных для хранения различных целых чисел, которая поддерживает 4 операции:

- «add x» — Добавить в структуру элемент  $x$  ( $1 \leq x \leq 10^9$ );
- «min» — Извлечь из структуры минимальный элемент;
- «max» — Извлечь из структуры максимальный элемент;
- «mid» — Извлечь из структуры медианный элемент (в случае чётности размера структуры следует вывести левый медианный элемент, то есть если в структуре хранится  $n$  элементов  $a_0 \leq a_1 \leq \dots \leq a_{n-1}$ , то процедура извлекает элемент  $a_{\frac{n-1}{2}}$ ).

### Формат входных данных

В первой строке входных данных записано единственное число  $M$  ( $1 \leq M \leq 200000$ ) — количество команд. В следующих строках дано по одной команде из тех, что идут выше.

### Формат выходных данных

На каждый запрос «min», «max» и «mid» выведите значение извлечённого элемента.

### Пример

стандартный ввод	стандартный вывод
6	1
add 3	2
add 1	3
add 2	
min	
mid	
mid	



## Задача О. Больше слияний

Имя входного файла: стандартный ввод  
Имя выходного файла: стандартный вывод  
Ограничение по времени: 5 секунд  
Ограничение по памяти: 256 мегабайт

Отсортируйте  $K$  отсортированных массивов в один очень большой отсортированный массив.

### Формат входных данных

В первой строке идет число  $K$  ( $1 \leq K \leq 1000$ ). Далее идет  $K$  блоков — описания массивов для слияния в следующем формате.

На первой строке идет число  $N_i$  — размер  $i$ -го массива. На второй строке через пробел идут  $N_i$  целых чисел, не превосходящих по модулю  $10^9$ .

Гарантируется, что

$$S := \sum_{i=1}^K N_i \leq 10^6$$

### Формат выходных данных

Выведите  $S$  целых чисел в порядке неубывания — массив, полученный в результате слияния  $K$  отсортированных массивов.

### Примеры

стандартный ввод	стандартный вывод
3 3 1 2 3 4 7 8 9 10 3 4 5 6	1 2 3 4 5 6 7 8 9 10
3 3 1 2 3 4 1 2 3 4 2 3 4	1 1 2 2 3 3 3 4 4

## Задача Р. Силовые поля

Имя входного файла: стандартный ввод  
Имя выходного файла: стандартный вывод  
Ограничение по времени: 3 секунды  
Ограничение по памяти: 256 мегабайт

В физико-биологической лаборатории исследуют воздействие излучения на растения при облучении через «силовые поля».

Экспериментальная установка содержит квадратную платформу размером  $10^9 \times 10^9$ , заполненную плодородной почвой. Над платформой установлен источник излучения. Между источником излучения и платформой можно включать  $n$  силовых полей.

Генератор силовых полей установлен над точкой  $(0, 0)$ . При этом  $i$ -е силовое поле представляет собой прямоугольник со сторонами, параллельными границам платформы и координатами двух противоположных углов  $(0, 0)$  и  $(x_i, y_i)$ .

В эксперименте планируется изучать воздействие излучения на растения при облучении через  $k$  силовых полей. Из заданных  $n$  полей необходимо выбрать  $k$  полей для эксперимента. Ученые хотят выбрать силовые поля таким образом, чтобы площадь участка платформы, над которой находятся все  $k$  выбранных силовых полей, была максимальна.

Требуется написать программу, которая по заданным целым числам  $n, k$  и описанию  $n$  силовых полей определяет, какие  $k$  силовых полей необходимо выбрать для эксперимента, чтобы площадь участка, покрытого всеми  $k$  силовыми полями, была максимальна, и выводит площадь этого участка.

### Формат входных данных

Первая строка входного файла содержит целые числа  $n$  и  $k$  ( $1 \leq k \leq n \leq 200000$ ) — общее количество силовых полей и количество силовых полей, которые необходимо выбрать для эксперимента.

Последующие  $n$  строк содержат по два целых числа  $x_i, y_i$  ( $1 \leq x_i, y_i \leq 10^9$ ) — координаты дальнего от начала координат угла прямоугольного участка  $i$ -го силового поля.

### Формат выходных данных

Требуется вывести одно целое число: максимальную площадь искомого участка.

### Пример

стандартный ввод	стандартный вывод
5 3 3 5 2 2 2 5 4 4 5 3	9

## Задача Q. Калькулятор

Имя входного файла:            стандартный ввод  
Имя выходного файла:        стандартный вывод  
Ограничение по времени:    1 секунда  
Ограничение по памяти:      256 мегабайт

Имеется калькулятор, который выполняет три операции:

1. Прибавить к числу  $X$  единицу;
2. Умножить число  $X$  на 2;
3. Умножить число  $X$  на 3.

Определите кратчайшую последовательность операций, необходимую для получения из числа 1 заданное число  $N$ .

### Формат входных данных

Программа получает на вход одно число  $N$ , не превосходящее  $10^6$ .

### Формат выходных данных

Требуется вывести одно число: наименьшее количество искомых операций.

### Примеры

стандартный ввод	стандартный вывод
1	0
5	3
962340	17

## Задача Р. Гвоздики

Имя входного файла: стандартный ввод  
Имя выходного файла: стандартный вывод  
Ограничение по времени: 1 секунда  
Ограничение по памяти: 256 мегабайт

В дощечке в один ряд вбиты гвоздики. Любые два гвоздика можно соединить ниточкой. Требуется соединить некоторые пары гвоздиков ниточками так, чтобы к каждому гвоздику была привязана хотя бы одна ниточка, а суммарная длина всех ниточек была минимальна.

### Формат входных данных

В первой строке входных данных записано число  $N$  — количество гвоздиков ( $2 \leq N \leq 100$ ). В следующей строке заданы  $N$  чисел — координаты всех гвоздиков (неотрицательные целые числа, не превосходящие 10000).

### Формат выходных данных

Выведите единственное число — минимальную суммарную длину всех ниточек.

### Пример

стандартный ввод	стандартный вывод
6 3 4 6 12 13 14	5

## Задача S. НВП

Имя входного файла:            стандартный ввод  
Имя выходного файла:        стандартный вывод  
Ограничение по времени:    1 секунда  
Ограничение по памяти:      256 мегабайт

Дана последовательность, требуется найти её наибольшую возрастающую подпоследовательность.

### Формат входных данных

В первой строке входных данных задано число  $N$  - длина последовательности ( $1 \leq N \leq 1000$ ). Во второй строке задается сама последовательность (разделитель — пробел). Элементы последовательности — целые числа, не превосходящие 10000 по модулю.

### Формат выходных данных

Требуется вывести наибольшую возрастающую подпоследовательность данной последовательности. Если таких подпоследовательностей несколько, необходимо вывести одну (любую) из них.

### Пример

стандартный ввод	стандартный вывод
10 4 8 2 6 2 10 6 29 58 9	4 8 10 29 58

## Задача Т. Быстрая НВП

Имя входного файла: стандартный ввод  
Имя выходного файла: стандартный вывод  
Ограничение по времени: 1 секунда  
Ограничение по памяти: 256 мегабайт

Дана последовательность, требуется найти длину её НВП.

### Формат входных данных

В первой строке входных данных задано число  $N$  — длина последовательности ( $1 \leq N \leq 10^5$ ).  
Во второй строке задается сама последовательность (разделитель — пробел). Элементы последовательности — целые числа, не превосходящие 100000 по модулю.

### Формат выходных данных

Длина НВП.

### Пример

стандартный ввод	стандартный вывод
10 -2 1 2 0 0 -2 -2 -2 1 -1	3

## Задача U. Число возрастающих подпоследовательностей

Имя входного файла: стандартный ввод  
Имя выходного файла: стандартный вывод  
Ограничение по времени: 1 секунда  
Ограничение по памяти: 256 мегабайт

Задана последовательность целых чисел. Найдите количество ее возрастающих подпоследовательностей.

### Формат входных данных

Первая строка входного файла содержит длину последовательности  $n$  ( $1 \leq n \leq 100$ ), а вторая — ее элементы (натуральные числа, меньшие 5000).

### Формат выходных данных

Выведите количество возрастающих подпоследовательностей по модулю  $10^6$ .

### Примеры

стандартный ввод	стандартный вывод
3 1 2 3	7
3 3 1 2	4

## Задача V. Рюкзак

Имя входного файла:            стандартный ввод  
Имя выходного файла:        стандартный вывод  
Ограничение по времени:    1 секунда  
Ограничение по памяти:      256 мегабайт

Дано  $N$  предметов массой  $m_1, \dots, m_N$  и стоимостью  $c_1, \dots, c_N$  соответственно. Ими наполняют рюкзак, который выдерживает вес не более  $M$ . Определите набор предметов, который можно унести в рюкзак, имеющий наибольшую стоимость.

### Формат входных данных

В первой строке вводится натуральное число  $N$ , не превышающее 100 и натуральное число  $M$ , не превышающее 10000.

Во второй строке вводятся  $N$  натуральных чисел  $m_i$ , не превышающих 100.

Во третьей строке вводятся  $N$  натуральных чисел  $c_i$ , не превышающих 100.

### Формат выходных данных

Выведите номера предметов (числа от 1 до  $N$ ), которые войдут в рюкзак наибольшей стоимости.

### Пример

стандартный ввод	стандартный вывод
4 6	1
2 4 1 2	3
7 2 5 1	4



## Задача W. НОП

Имя входного файла: стандартный ввод  
Имя выходного файла: стандартный вывод  
Ограничение по времени: 1 секунда  
Ограничение по памяти: 256 мегабайт

Даны две последовательности, требуется найти и вывести их наибольшую общую подпоследовательность.

### Формат входных данных

В первой строке входных данных содержится число  $N$  — длина первой последовательности ( $1 \leq N \leq 1000$ ). Во второй строке заданы члены первой последовательности (через пробел) — целые числа, не превосходящие 10000 по модулю.

В третьей строке записано число  $M$  — длина второй последовательности ( $1 \leq M \leq 1000$ ). В четвертой строке задаются члены второй последовательности (через пробел) — целые числа, не превосходящие 10000 по модулю.

### Формат выходных данных

Требуется вывести наибольшую общую подпоследовательность данных последовательностей, через пробел.

### Пример

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

## Задача X. Редакторское расстояние

Имя входного файла:            стандартный ввод  
Имя выходного файла:        стандартный вывод  
Ограничение по времени:    1 секунда  
Ограничение по памяти:      256 мегабайт

Дана текстовая строка. С ней можно выполнять следующие операции:

- Заменить один символ строки на другой символ.
- Удалить один произвольный символ.
- Вставить произвольный символ в произвольное место строки.

Например, при помощи первой операции из строки «СОК» можно получить строку «СУК», при помощи второй операции — строку «ОК», при помощи третьей операции — строку «СТОК». Минимальное количество таких операций, при помощи которых можно из одной строки получить другую, называется стоимостью редактирования или расстоянием Левенштейна. Определите расстояние Левенштейна для двух данных строк.

### Формат входных данных

Программа получает на вход две строки, длина каждой из которых не превосходит 5000 символов, строки состоят только из заглавных латинских букв.

### Формат выходных данных

Требуется вывести одно число — расстояние Левенштейна для данных строк.

### Пример

стандартный ввод	стандартный вывод
ABCDEFGH ACDEXGIN	3

## Задача Y. Дешифровка

Имя входного файла: стандартный ввод  
Имя выходного файла: стандартный вывод  
Ограничение по времени: 1 секунда  
Ограничение по памяти: 512 мегабайт

Алиса и Боб придумали очередной способ кодировать свои послания друг другу. Алиса посылает Бобу строку  $S$ , а Боб ищет ключ, чтобы расшифровать сообщение. Модуль дешифровки уже написан, а вот помочь Бобу с поиском ключа по сообщению — ваша задача.

Алиса присылает Бобу строку  $S$ , ключ — натуральное число, равное минимальному количеству символов, которое надо удалить из  $S$ , чтобы получился палиндром. Ваша задача — найти по строке  $S$  искомый ключ.

### Формат входных данных

Дана строка  $S$  из строчных латинских букв, ее длина не превышает  $10^4$ .

### Формат выходных данных

Выведите единственное число — ключ для дешифровки сообщения.

### Примеры

стандартный ввод	стандартный вывод
abac	1
abacaba	0

## Задача Z. Построение

Имя входного файла:	стандартный ввод
Имя выходного файла:	стандартный вывод
Ограничение по времени:	1 секунда
Ограничение по памяти:	256 мегабайт

На построение по случаю прилёта генерала собрался весь личный состав космической военной части. Солдаты выстроили свои корабли в  $X$  рядов по  $Y$  кораблей. У каждого корабля есть некоторое количество знаков отличия.

Генерал стартует с точки  $(0, 0)$ . Далее он ищет корабль, имеющий наибольшее количество знаков отличия, и перелетает к нему. Генерал делает такие перелёты, пока не окажется в ситуации, что дальнейший перелет совершить невозможно (так как он останется на месте при попытке полететь дальше). Проблема заключается в том, что угол обзора генерала ограничен, а именно, стоя у корабля с координатами  $(0, 0)$  (то есть около левого в нулевом ряду), он видит только корабли с координатами  $(a, b)$ , где  $a \in [0, h), b \in [0, w)$ . Точно так же, находясь у корабля  $(i, j)$ , он видит корабли с координатами  $(a, b)$ , где  $a \in [i, i + h), b \in [j, j + w)$ .

Необходимо вывести координаты корабля, к которому в конечном итоге подлетит генерал. Все корабли имеют разное количество знаков отличия.

### Формат входных данных

В первой строке идут два целых числа  $X, Y$  — размеры строя ( $1 \leq X \cdot Y \leq 10^6$ ).

Во второй строке идут два целых числа  $h, w$  — область видимости генерала ( $1 \leq h \leq X, 1 \leq w \leq Y$ ).

Далее идут  $X$  строк по  $Y$  целых чисел  $s_{ij}$  — количество знаков отличия у кораблей. ( $0 \leq s_{ij} \leq 10^9$ ). Гарантируется, что все  $s_{ij}$  различны.

Оси координат заданы так, что корабль с координатами  $(0, 0)$  идет первым в первой строке описания количества наград у кораблей, а корабль с координатами  $(0, Y - 1)$  идет последним в той же строке.

### Формат выходных данных

В ответе необходимо вывести два числа через пробел — координаты корабля, у которого остановится генерал в результате таких перелетов.

### Примеры

стандартный ввод	стандартный вывод
2 2 1 2 3 4 1 2	0 1
2 2 2 1 1 2 3 4	1 0
2 2 2 2 1 2 3 4	1 1

## Задача ЗА. Дуэль

Имя входного файла:	стандартный ввод
Имя выходного файла:	стандартный вывод
Ограничение по времени:	1 секунда
Ограничение по памяти:	256 мегабайт

Отряд Дамблдора готовится к полёту в Отдел Тайн Министерства Магии. Так как на их пути за Пророчеством могут встретиться слуги Волан-де-Морта, Гарри с Невиллом решили потренироваться в дуэльных боях.

Они решили провести  $n$  раундов дуэли, а после каждого раунда выпивать восстанавливающее зелье, чтобы не тратить лишних сил. Всего у ребят  $2n$  зелий. После каждого раунда дуэли они выбирают два наиболее близких по мощности восстанавливающих зелья.

Из нескольких таких пар они выбирают ту, у которой сумма мощностей наибольшая. Затем Невилл выпивает более мощное зелье из пары, а Гарри — менее мощное.

Помогите Гарри с Невиллом определить, какие зелья и в каком порядке выпьет каждый из них.

### Формат входных данных

Первая строка входного файла содержит одно целое число  $n$  ( $1 \leq n \leq 100000$ ) — количество раундов дуэли. Во второй строке входного файла содержится  $2n$  целых чисел  $a_1, a_2, \dots, a_{2n}$  ( $1 \leq a_i \leq 10^9$ ), где  $a_i$  — мощность  $i$ -го восстанавливающего зелья.

### Формат выходных данных

В первой строке выведите  $n$  чисел — мощности зелий, которые выпьет Невилл. Во второй строке выведите  $n$  чисел — мощности зелий, которые выпьет Гарри. Зелья надо выводить в том порядке, в котором Невилл и Гарри должны их выпивать.

### Примеры

стандартный ввод	стандартный вывод
1 3 6	6 3
2 11 43 56 22	22 56 11 43

## Задача ZB. Менеджер памяти

Имя входного файла:	стандартный ввод
Имя выходного файла:	стандартный вывод
Ограничение по времени:	1 секунда
Ограничение по памяти:	256 мегабайт

Пете поручили написать менеджер памяти для новой стандартной библиотеки языка C++. В распоряжении у менеджера находится массив из  $N$  последовательных ячеек памяти, пронумерованных от 1 до  $N$ . Задача менеджера — обрабатывать запросы приложений на выделение и освобождение памяти.

Запрос на выделение памяти имеет один параметр  $K$ . Такой запрос означает, что приложение просит выделить ему  $K$  последовательных ячеек памяти. Если в распоряжении менеджера есть хотя бы один свободный блок из  $K$  последовательных ячеек, то он обязан в ответ на запрос выделить такой блок. При этом непосредственно перед самой первой ячейкой памяти выделяемого блока не должно располагаться свободной ячейки памяти. После этого выделенные ячейки становятся занятыми и не могут быть использованы для выделения памяти, пока не будут освобождены. Если блока из  $K$  последовательных свободных ячеек нет, то запрос отклоняется.

Запрос на освобождение памяти имеет один параметр  $T$ . Такой запрос означает, что менеджер должен освободить память, выделенную ранее при обработке запроса с порядковым номером  $T$ . Запросы нумеруются, начиная с единицы. Гарантируется, что запрос с номером  $T$  — запрос на выделение, причем к нему еще не применялось освобождение памяти. Освобожденные ячейки могут снова быть использованы для выделения памяти. Если запрос с номером  $T$  был отклонен, то текущий запрос на освобождение памяти игнорируется.

Требуется написать менеджер памяти, удовлетворяющий приведенным критериям.

### Формат входных данных

Первая строка входного файла содержит числа  $N$  и  $M$  — количество ячеек памяти и количество запросов соответственно ( $1 \leq N \leq 2^{31} - 1$ ;  $1 \leq M \leq 10^5$ ). Каждая из следующих  $M$  строк содержит по одному числу:  $(i + 1)$ -я строка входного файла ( $1 \leq i \leq M$ ) содержит либо положительное число  $K$ , если  $i$ -й запрос — запрос на выделение с параметром  $K$  ( $1 \leq K \leq N$ ), либо отрицательное число  $-T$ , если  $i$ -й запрос — запрос на освобождение с параметром  $T$  ( $1 \leq T \leq i$ ).

### Формат выходных данных

Для каждого запроса на выделение памяти выведите в выходной файл результат обработки этого запроса: для успешных запросов выведите номер первой ячейки памяти в выделенном блоке, для отклоненных запросов выведите число «-1». Результаты нужно выводить в порядке следования запросов во входном файле.

### Пример

стандартный ввод	стандартный вывод
6 8	1
2	3
3	-1
-1	-1
3	1
3	-1
-5	
2	
2	

## Задача ZС. Динамический массив

Имя входного файла: стандартный ввод  
Имя выходного файла: стандартный вывод  
Ограничение по времени: 1 секунда  
Ограничение по памяти: 256 мегабайт

В этой задаче вам предстоит написать динамический массив, который умеет обрабатывать следующие запросы:

- «push  $n$ » — добавить в конец элемент  $n$  ( $-10^9 \leq n \leq 10^9$ ), программа должна вывести «ok»;
- «pop» — удалить элемент из конца, программа должна вывести его значение. Если массив пустой, то вывести «error»;
- «get  $k$ » — вывести элемент с индексом  $k$  в 0-индексации. Если индекс больше размера массива, то вывести строку «Incorrect index»;
- «size» — вывести размер массива;
- «clear» — очистить массив, программа должна вывести «ok».

Запрещено выделять память заведомо под размер количества команд. Ваша структура должна работать для произвольного количества элементов.

### Формат входных данных

В первой строке идет количество команд  $N$  ( $1 \leq N \leq 10^5$ ). Далее идет  $N$  команд в формате выше по одной на строку.

### Формат выходных данных

Вывести результаты обработки команд.

### Пример

стандартный ввод	стандартный вывод
17	error
pop	0
size	ok
clear	0
size	ok
push 1	1
get 0	Incorrect index
get 1	1
size	1
pop	ok
clear	ok
push 0	ok
push 0	0
get 1	2
size	0
pop	1
size	0
pop	

## Задача ZD. НОВП

Имя входного файла:            стандартный ввод  
Имя выходного файла:        стандартный вывод  
Ограничение по времени:    1 секунда  
Ограничение по памяти:      256 мегабайт

### Формат входных данных

Во входном файле записаны две последовательности. Каждая последовательность описывается двумя строками следующим образом: в первой строке идет длина последовательности  $M$  ( $1 \leq M \leq 500$ ), во второй идут  $M$  целых чисел  $a_i$  ( $-2^{31} \leq a_i \leq 2^{31} - 1$ ) — члены последовательности.

### Формат выходных данных

В первой строке выходного файла выведите  $N$  — длину наибольшей общей возрастающей подпоследовательности. Во второй строке выходного файла выведите саму подпоследовательность.

### Пример

стандартный ввод	стандартный вывод
5 1 4 2 5 -12 4 -12 1 2 4	2 1 4