

## Задача А. Минимум на отрезке

Имя входного файла: стандартный ввод  
Имя выходного файла: стандартный вывод  
Ограничение по времени: 0.5 секунд  
Ограничение по памяти: 256 мегабайт

Рассмотрим последовательность целых чисел длины  $N$ . По ней с шагом 1 двигается «окно» длины  $K$ , то есть сначала в «окне» видно первые  $K$  чисел, на следующем шаге в «окне» уже будут находиться  $K$  чисел, начиная со второго, и так далее до конца последовательности. Требуется для каждого положения «окна» определить минимум в нём.

### Формат входных данных

В первой строке входных данных содержатся два числа  $N$  и  $K$  ( $1 \leq N \leq 150000$ ,  $1 \leq K \leq 10000$ ,  $K \leq N$ ) – длины последовательности и «окна», соответственно. На следующей строке находятся  $N$  чисел – сама последовательность. Числа последовательности не превосходят по модулю  $10^5$ .

### Формат выходных данных

Выходные данные должны содержать  $N - K + 1$  строк – минимумы для каждого положения «окна».

### Пример

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

## Задача В. Словарь синонимов

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

Вам дан словарь, состоящий из пар слов. Каждое слово является синонимом к парному ему слову. Все слова в словаре различны. Для одного данного слова определите его синоним.

### Формат входных данных

Программа получает на вход количество пар синонимов  $N \leq 100000$ . Далее следует  $N$  строк, каждая строка содержит ровно два слова-синонима. После этого следует одно слово.

### Формат выходных данных

Программа должна вывести синоним к данному слову.

### Пример

стандартный ввод	стандартный вывод
3 Hello Hi Bye Goodbye List Array Goodbye	Bye

### Замечание

Используйте для решения средства стандартной библиотеки!

## Задача С. Вычислительная ихтиология

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

Игорь работает младшим лаборантом в НИИ ихтиологии. Ему вверены  $n$  аквариумов, стоящих в ряд, в каждом из которых живет колония рыбок гуппи. Про каждую колонию заранее известна ее численность.

В лабораторных условиях НИИ ихтиологии колония рыбок гуппи растет по следующему правилу: достигнув популяции в  $f$  рыбок, колония живет в течение  $\max(1000 - f, 1)$  секунд, после чего на свет появляется новая рыбка. От начального момента времени до рождения первой рыбки колония размера  $f$  также ждет  $\max(1000 - f, 1)$  секунд.

Например, колония с начальным размером 996 будет размножаться следующим образом:

время	число рыб	время до очередной рыбки
0	996	4
4	997	3
7	998	2
9	999	1
10	1000	1
11	1001	1
...	...	...

Появление на свет каждой новой рыбки Игорь должен фиксировать в специальном журнале. Будем считать, что запись он делает мгновенно, но при этом он должен в момент рождения новой рыбки находиться рядом с аквариумом, в котором это произошло.

На перемещение от одного аквариума к соседнему у Игоря уходит одна секунда. В начальный момент времени Игорь стоит около первого аквариума.

Вычислите, в течение какого наибольшего периода времени Игорь сможет добросовестно выполнять свою работу.

### Формат входных данных

В первой строке входного файла содержится целое число  $n$  ( $2 \leq n \leq 50$ ) — количество аквариумов с рыбками гуппи в НИИ ихтиологии. Каждая из следующих  $n$  строк содержит одно целое число  $a_i$  ( $1 \leq a_i \leq 2007$ ) — численность  $i$ -й колонии.

### Формат выходных данных

В выходной файл выведите момент времени, когда родится первая рыбка гуппи, запись о рождении которой Игорь сделать не сможет.

### Пример

стандартный ввод	стандартный вывод
3 996 1 994	7

### Замечание

В приведенном примере Игорь сначала ждет у первого аквариума появления рыбки на 4-й секунде. После этого он бежит к третьему аквариуму (на это у него уходит 2 секунды) и как раз успевает к рождению рыбки на 6-й секунде. Однако вернуться к первому аквариуму, где следующая рыбка родится на 7-й секунде, он уже не успевает.

## Задача D. Множества

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

На вступительном констесте в пилотную группу по программированию Вашему другу предложили реализовать структуру данных для хранения множеств чисел. Так как он специализируется на истории литературы, данную структуру придётся реализовать Вам.

Структура должна хранить  $m + 1$  множеств чисел от 0 до  $n$ , пронумерованных от 0 до  $m$  включительно, при этом одно число может принадлежать сразу нескольким множествам. Изначально все множества пустые.

Вы должны реализовать следующие операции на этой структуре:

1. ADD  $e$   $s$

Добавить в множество  $N^s$  ( $0 \leq s \leq m$ ) число  $e$  ( $0 \leq e \leq n$ ).

2. DELETE  $e$   $s$

Удалить из множества  $N^s$  ( $0 \leq s \leq m$ ) число  $e$  ( $0 \leq e \leq n$ ). Гарантируется, что до этого число  $e$  было помещено в множество

3. CLEAR  $s$

Очистить множество  $N^s$  ( $0 \leq s \leq m$ ).

4. LISTSET  $s$

Показать содержимое множества  $N^s$  ( $0 \leq s \leq m$ ) **в возрастающем порядке**, либо  $-1$ , если множество пусто.

5. LISTSETSOF  $e$

Вывести номера множеств, в которых лежит число  $e$  ( $0 \leq e \leq n$ ) **в возрастающем порядке**, либо  $-1$ , если этого числа нет ни в одном множестве.

### Формат входных данных

Сначала вводятся числа  $N$  ( $1 \leq N \leq 9223372036854775807$ ),  $M$  ( $1 \leq M \leq 100000$ ) и  $K$  ( $0 \leq K \leq 100000$ ) — максимальное число, номер максимального множества и количество запросов к структуре данных. Далее следуют  $K$  строк указанного формата запросов.

### Формат выходных данных

На каждый запрос LISTSET Ваша программа должна вывести числа — содержимое запрошенного множества или  $-1$ , если множество пусто.

На каждый запрос LISTSETSOF Ваша программа должна вывести числа — номера множеств, содержащих запрошенное число, или  $-1$ , если таких множеств не существует.

На прочие запросы не должно быть никакого вывода.

Гарантируется, что правильный вывод программы не превышает одного мегабайта.

## Пример

стандартный ввод	стандартный вывод
10 10	1 2
9	1 2
ADD 1 1	2
ADD 1 2	-1
ADD 2 1	
LISTSET 1	
LISTSETSOFF 1	
DELETE 1 1	
LISTSET 1	
CLEAR 1	
LISTSET 1	

## Замечание

Эту задачу можно (и нужно!) решать, используя `std::set` и `std::map`.

## Задача E. T2005

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

Клавиатура сотового телефона выглядит так:

1 пробел	2 abc	3 def
4 ghi	5 jkl	6 mno
7 pqrs	8 tuv	9 wxyz

Режим ввода T2005 устроен следующим образом. В телефоне есть словарь. Пользователь, чтобы ввести слово, последовательно нажимает клавиши, на которых написаны буквы этого слова. Например, чтобы ввести слово `begin` пользователь должен нажимать клавиши `23446`. Но как только в словаре оказывается только одно слово с таким началом, это слово автоматически подставляется и, кроме того, после этого слова автоматически добавляется пробел. Например, пусть пользователь нажал клавиши `234`, и оказалось, что слов, ввод которых начинается с нажатия именно этих клавиш, — ровно одно. Тогда автоматически подставится это слово и пробел после него, а все последующие нажатия клавиш уже будут относиться к вводу следующего слова.

Если для ввода какого-то слова нужно нажать последовательность клавиш, которая может являться началом какого-то другого слова, то после ввода этого слова нужно нажать клавишу `1`, что соответствует вводу пробела. При вводе пробела считается, что вы ввели все слово целиком (а не только какое-либо его начало). Если после ввода пробела оказалось, что в словаре такой последовательности клавиш удовлетворяет несколько слов, подставляется первое из них в алфавитном порядке. Если (опять же после ввода пробела) оказалось, что в словаре нет слова, которое может быть введено такой последовательностью клавиш, то все, что было введено после предыдущего пробела (введенного или автоматически подставленного, или, если в тексте ранее не встречалось ни одного пробела — от начала текста) удаляется. Если после ввода пробела (как нажатием «1», так и автоподстановкой) или в начале текста нажимается клавиша «1», то ее нажатие игнорируется.

Вам дан словарь и последовательность нажатий клавиш. Выведите текст, который был введен пользователем.

Примечание: в тексте используются только маленькие латинские буквы и символ пробел.

### Формат входных данных

Сначала на вход программы поступает число  $N$  — количество слов в словаре ( $2 \leq N \leq 100000$ ). В следующих  $N$  строках задается словарь. Каждое слово записано в отдельной строке. Слова расположены в алфавитном порядке. Никакое слово в словаре не встречается дважды. Длина каждого слова не превосходит 10 символов.

Далее вводится число  $M$  — количество нажатий клавиш ( $1 \leq M \leq 20000$ ). Затем задается  $M$  разделяющихся пробелами чисел, описывающих нажатые клавиши. Последней нажатой клавишей всегда является клавиша «1».

### Формат выходных данных

Выведите одну строку — текст, который оказался введен пользователем. Пробел после последнего введенного слова также должен быть выведен.

## Пример

стандартный ввод	стандартный вывод
5 po pod sasha shla shosse 12 7 4 5 7 2 7 6 1 7 4 6 1	shla sasha po shosse

## Задача F. Тетрис

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

Как и в обычном тетрисе, поле в игре Strategy Tetris представляет собой «стакан» шириной в  $W$  клеток ( $1 \leq W \leq 10^9$ ) и бесконечной высоты. В этот стакан падают сверху  $N$  фигурок ( $1 \leq N \leq 100000$ ).  $i$ -я фигурка представляет собой прямоугольник шириной в  $W_i$  клеток и высотой в одну клетку; самая левая клетка фигурки имеет абсциссу  $a_i$  ( $1 \leq a_i \leq W - W_i + 1$ ). Фигурки падают по обычным правилам: если при падении фигурка хотя бы одной своей клеткой ложится на какую-либо уже упавшую фигурку, то ее движение прекращается.

В отличие от обычного тетриса, игрок не имеет возможности вращать фигурки или смещать их по горизонтали в процессе падения — еще бы, это пришлось бы делать быстро и не было бы времени серьёзно подумать над стратегией. Единственное, что он может — это выбрать порядок, в котором эти  $N$  фигурок упадут в стакан (каждая по одному разу). Ваша задача — помочь ему выбрать такой порядок, при котором высота образовавшейся в результате падения конструкции была бы как можно меньше. (В отличие от обычного тетриса, полностью заполненная фигурками горизонталь никуда не исчезает).

На рисунке ниже приведен пример заполнения стакана фигурками из примера входных данных (порядок заполнения соответствует выходному файлу, приведенному в примере).

	2		
1		3	

### Формат входных данных

В первой строке входного файла записаны числа  $N$  и  $W$ , а в последующих  $N$  строках — пары чисел  $a_i$  и  $W_i$ .

### Формат выходных данных

Выведите в выходной файл минимальную возможную высоту конструкции, а затем последовательность номеров фигурок, к этой высоте приводящую. Фигурки нумеруются натуральными числами от 1 до  $N$  в том порядке, в котором они указаны во входных данных. Если возможных вариантов несколько, выведите любой из них.

### Пример

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



## Задача G. 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

В первой строке входных данных задаются 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 входных данных ( $1 \leq i \leq M$ ) 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  $T$  ( $1 \leq T < i$ ).

### Формат выходных данных

Для каждого запроса на выделение памяти выведите 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 входных данных

### Пример

стандартный ввод	стандартный вывод
42 9	1
7	8
3	11
8	19
-2	25
6	30
5	19
-5	
9	
4	

## Задача Н. Машины

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

Петя, которому три года, очень любит играть с машинками. Всего у Пети  $N$  различных машинок, которые хранятся на полке шкафа так высоко, что он сам не может до них дотянуться. Одновременно на полу комнаты может находиться не более  $K$  машинок. Петя играет с одной из машинок на полу и если он хочет поиграть с другой машинкой, которая также находится на полу, то дотягивается до нее сам. Если же машинка находится на полке, то он обращается за помощью к маме. Мама может достать для Пети машинку с полки и одновременно с этим поставить на полку любую машинку с пола. Мама очень хорошо знает своего ребенка и может предугадать последовательность, в которой Петя захочет играть с машинками. При этом, чтобы не мешать Петиней игре, она хочет совершить как можно меньше операций по подъему машинки с пола, каждый раз правильно выбирая машинку, которую следует убрать на полку. Ваша задача состоит в том, чтобы определить минимальное количество операций. Перед тем, как Петя начал играть, все машинки стоят на полке.

### Формат входных данных

В первой строке содержатся три числа  $N$ ,  $K$  и  $P$  ( $1 \leq K, N \leq 100000$ ,  $1 \leq P \leq 500000$ ). В следующих  $P$  строках записаны номера машинок в том порядке, в котором Петя захочет играть с ними.

### Формат выходных данных

Выведите единственное число: минимальное количество операций, которое надо совершить Петиней маме.

### Пример

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

### Замечание

В этой задаче можно использовать STL.

Пояснения к примеру:

Операция 1: снять машинку 1

Операция 2: снять машинку 2

Операция 3: поднять машинку 2 и снять машинку 3

Операция 4: поднять машинку 3 или 1 и снять машинку 2