

А. Отрезки

ограничение по времени на тест: 4 секунды

ограничение по памяти на тест: 256 мегабайт

ввод: стандартный ввод

вывод: стандартный вывод

Дано m отрезков чисел от 1 до n . Для каждого числа посчитайте количество накрывающих его отрезков.

Входные данные

Первая строка содержит два целых числа n, m ($1 \leq n, m \leq 10^6$) — размер множества, на котором рассматриваются отрезки и число отрезков.

Каждая из следующих m строк содержит два числа l и r ($1 \leq l \leq r \leq n$) — очередной отрезок.

Выходные данные

Выведите n чисел, k -е из которых — число отрезков, накрывающих число k .

Примеры

входные данные	Сору
6 3 2 4 1 3 2 5	
выходные данные	Сору
1 3 3 2 1 0	

входные данные	Сору
10 3 6 10 1 10 2 2	
выходные данные	Сору
1 2 1 1 1 2 2 2 2 2	

В. Турнирная таблица индивидуальных

ограничение по времени на тест: 10 секунд
ограничение по памяти на тест: 256 мегабайт
ввод: стандартный ввод
вывод: стандартный вывод

В индивидуальном первенстве по Дартс, участникам необходимо было набрать максимальное число очков за попытку. Количество попыток не ограничивалось. В протокол комиссии попадало имя участника и результат за проделанную попытку. Итоговым результатом участника считалась лучшая попытка. Если результаты совпадают, первым считается игрок показавший данный результат раньше. Помогите организаторам выстроить участников по убыванию.

Входные данные

Сначала задано число участников M ($1 \leq M \leq 100$), затем число попыток всех участников N ($1 \leq N \leq 10^5$). Далее идёт N , в каждой из которой записано имя участника и набранные им очки. Набранные очки — целое положительное число, имя участника — строка не длиннее 7 символов, состоящая из строчных английских букв и цифр.

Участник не может набрать больше 10^9 очков за одну попытку.

Выходные данные

M строк с именами участников в порядке убывания занятых мест.

Пример

входные данные	Copy
<pre>4 6 Ivan 2 Boris 4 Ivan 4 Boris 2 Petr 4 David 8</pre>	
выходные данные	Copy
<pre>David Boris Ivan Petr</pre>	

Примечание

При решении задачи реализуйте структуру данных куча или очередь с приоритетами на основе кучи. Структура данных должна быть выделена в отдельный класс и должны быть реализованы стандартные методы: `insert` и `extract_max` (или `extract_min`), а в случае очереди с приоритетами также операция изменения приоритета

В случае, если вместо собственной реализации будет использована куча из стандартной библиотеки C++ или библиотеки Питона, или же решение будет без использования кучи, задача оценивается в половину баллов.

С. Камера хранения

ограничение по времени на тест: 5 секунд

ограничение по памяти на тест: 256 мегабайт

ввод: стандартный ввод

вывод: стандартный вывод

В камере хранения ***M*** ячеек. Желая сдать багаж сообщает своё имя и время когда он должен забрать багаж, и он обязательно его заберёт. Если в камере хранения нет свободных ячеек, ему отказывают, а если есть места, то обязательно принимают багаж.

В камере хранения необходимо вести журнал. Если у посетителя принимают багаж, то в журнал записывается его имя, если не принимают, в журнал записывают «No». Даже если один и тот же посетитель пришёл два раза, то он обязан получить новую ячейку, а не обязательно воспользоваться старой. На выходе должно быть записано содержание журнала.

Все посетители приходят последовательно, то есть если первый посетитель пришёл в 9:00, то второй не может прийти раньше, скажем, в 8:59. Камера хранения работает только один день, поэтому время задано в границах 0:00 - 23:59.

В случае, когда в одну и ту же минуту должны сдать багаж и забрать багаж, считается, что сдающий пришёл раньше забирающего (если камера хранения была заполнена и должна была освободиться, то сдающий получит отказ)

Входные данные

На вход подаётся число посетителей $N \leq 10^6$ и число ячеек $M \leq 10^5$. После чего идёт ***N*** строк с указанием имени сдающего, времени прихода и времени выдачи багажа. Имя сдающего состоит из не более, чем **10** символов и может содержать строчные и прописные английские буквы, а также цифры. Время задано в формате h:mm если ***h*** < **10** или hh:mm в противном случае. В качестве разделителя используется одинарный пробел.

Выходные данные

N строк. В ***i***-й строке стоит имя ***i***-го посетителя, если у него приняли багаж и «No», если не приняли багаж.

Пример

входные данные	Сору
<pre>4 2 Ivan 0:04 4:58 Denis 0:12 13:52 Pavel 0:21 15:52 Alex 6:41 19:30</pre>	
выходные данные	Сору
<pre>Ivan Denis No Alex</pre>	

D. Свaпнем

ограничение по времени на тест: 2 секунды

ограничение по памяти на тест: 64 мегабайта

ввод: стандартный ввод

вывод: стандартный вывод

Все положительные целые числа записали в бесконечный массив: $N[i] = i, i \geq 1$. С массивом разрешено выполнять операцию **swap**(a, b), которая меняет элементы массива со значениями a и b местами, а в качестве результата возвращает расстояние между этими элементами — модуль разности их индексов.

Нужно для последовательности операций **swap** на входе записать последовательность значений операций.

Входные данные

n — количество операций $1 \leq n \leq 200\,000$.

Далее идёт n строк с аргументами операций **swap**:

$a_i\ b_i$ ($a_i, b_i \in \{1, \dots, 10^9\}$)

Выходные данные

Результаты операций по одной на каждую строку.

Пример

входные данные	Сору
4 1 4 1 3 4 5 1 4	
выходные данные	Сору
3 1 4 2	

Е. Метод Д'Ондта

ограничение по времени на тест: 1 секунда

ограничение по памяти на тест: 256 мегабайт

ввод: стандартный ввод

вывод: стандартный вывод

В Берляндии недавно прошли парламентские выборы, в которых приняло участие n партий, которые набрали v_1, \dots, v_n голосов соответственно. Для распределения мест между партиями используется следующий механизм, максимизирующий количество голосов, приходящихся на одно выделенное место:

1. Для каждой партии вычисляется квота, равная $\frac{v_k}{s_k+1}$, где s_k — количество мест, которые уже были выделены для партии (изначально $s_k = 0$).
2. Партии с наибольшей квотой присваивается одно новое место. В случае равенства квот место присваивается партии с наименьшим номером.
3. Процедура продолжается пока не будут распределены все m мест в берляндском парламенте.

Вам необходимо провести распределение мест между партиями.

Входные данные

В первой строке содержатся два числа n и m ($1 \leq n, m \leq 10^5$) — количество партий в Берляндии и количество мест в Берляндском парламенте соответственно.

Во второй строке содержится n чисел v_1, \dots, v_n ($0 \leq v_i \leq 10^9$) — количество голосов, полученных каждой партией на выборах.

Выходные данные

Выведите n чисел s_1, \dots, s_n — количество мест, выделенной каждой партии в парламенте по результатам выборов.

Примеры

входные данные	Сору
3 10 50000 42000 19000	
выходные данные	Сору
5 4 1	

входные данные	Сору
5 8 1 1 1 1 1	
выходные данные	Сору
2 2 2 1 1	

входные данные	Сору
2 10 100 1	
выходные данные	Сору
10 0	

F. Same Differences

time limit per test: 2 seconds

memory limit per test: 256 megabytes

input: standard input

output: standard output

You are given an array a of n integers. Count the number of pairs of indices (i, j) such that $i < j$ and $a_j - a_i = j - i$.

Input

The first line contains one integer t ($1 \leq t \leq 10^4$). Then t test cases follow.

The first line of each test case contains one integer n ($1 \leq n \leq 2 \cdot 10^5$).

The second line of each test case contains n integers a_1, a_2, \dots, a_n ($1 \leq a_i \leq n$) — array a .

It is guaranteed that the sum of n over all test cases does not exceed $2 \cdot 10^5$.

Output

For each test case output the number of pairs of indices (i, j) such that $i < j$ and $a_j - a_i = j - i$.

Example

input

Copy

```
4
6
3 5 1 4 6 6
3
1 2 3
4
1 3 3 4
6
1 6 3 4 5 6
```

output

Copy

```
1
3
3
10
```


Г. Подотрезки перестановок

ограничение по времени на тест: 1 s.

ограничение по памяти на тест: 256 MB

ввод: standard input

вывод: standard output

Подотрезок перестановки считается интересным, если ни один из элементов на его концах не является ни минимумом, ни максимумом всего подотрезка. Вам потребуется найти такой подотрезок у перестановки. Более формально, вас просят найти такие числа l и r ($1 \leq l \leq r \leq n$), что $a_l \neq \min(a_l, a_{l+1}, \dots, a_r)$, $a_l \neq \max(a_l, a_{l+1}, \dots, a_r)$ и $a_r \neq \min(a_l, a_{l+1}, \dots, a_r)$, $a_r \neq \max(a_l, a_{l+1}, \dots, a_r)$.

Перестановкой длины n называется массив, состоящий из n различных целых чисел от 1 до n в произвольном порядке. Например, $[2, 3, 1, 5, 4]$ является перестановкой, но $[1, 2, 2]$ не является перестановкой (2 встречается дважды в массиве) и $[1, 3, 4]$ тоже не является перестановкой ($n = 3$, но 4 присутствует в массиве).

Найдите такой подотрезок, либо скажите, что такого подотрезка не существует.

Входные данные

Каждый тест состоит из нескольких наборов входных данных. Первая строка содержит единственное число t ($1 \leq t \leq 10^4$) — количество наборов входных данных. Далее следует описание наборов входных данных.

Для каждого набора входных данных в первой строке содержится одно целое число одно число n ($1 \leq n \leq 2 \cdot 10^5$) — длина перестановки.

Вторая строка содержит n различных целых чисел a_1, a_2, \dots, a_n ($1 \leq a_i \leq n$) — элементы перестановки.

Гарантируется, что сумма n по всем наборам входных данных не превосходит $2 \cdot 10^5$.

Выходные данные

Для каждого набора входных данных выведите -1 , если искомого подотрезка не существует.

Иначе, выведите два индекса l, r , такие что $[a_l, a_{l+1}, \dots, a_r]$ удовлетворяет всем условиям.

Если существует несколько решений, то выведите любое из них.

Пример

входные данные	Сору
4 3 1 2 3 4 2 1 4 3 7 1 3 2 4 6 5 7 6 2 3 6 5 4 1	
выходные данные	Сору
-1 1 4 2 6 -1	

Примечание

В первом и четвертом наборе входных данных можно показать, что искомым отрезков нет.

Во втором наборе входных данных подотрезок $[1, 4]$ удовлетворяет всем условиям, потому что $\max(a_1, a_2, a_3, a_4) = 4$, $\min(a_1, a_2, a_3, a_4) = 1$, и как мы видим, все условия выполняются.

В третьем наборе входных данных подотрезок $[2, 6]$ также удовлетворяет всем описанным условиям.