

Игра для класса

Одна из известных средних школ Ташкента отмечает свой юбилей, проводя игру между учениками и учителями. В классе сидят N учеников, пронумерованных от 0 до $N - 1$. Каждый ученик держит лист бумаги, на котором записан массив целых чисел. Изначально каждый лист пустой, поэтому на нем находится пустой массив.

Ученики играют в игру с M учителями, пронумерованными от 0 до $M - 1$, и директором школы. Игра состоит из M раундов, также пронумерованных от 0 до $M - 1$. На раунде j учитель j входит в класс, и происходят следующие события:

- Некоторые (возможно, ни одного) ученики поднимают руку. Гарантируется, что на каждом раунде хотя бы один ученик **не** поднимает руку, и каждый ученик может поднять руку **не более одного раза** за всю игру.
- Учитель просматривает листы, которые в данный момент находятся у всех учеников, и может **изменить** листы тех учеников, которые **не** подняли руку на этом раунде. Для каждого такого ученика учитель может заменить содержимое его листа новым (возможно, пустым) массивом целых чисел. Каждое целое число должно быть в диапазоне от 0 до 63 включительно, и массив может содержать не более 63 элементов.
- После этих изменений учитель j уходит, и ученики обмениваются своими листами в соответствии с секретной перестановкой P . А именно, для всех i ($0 \leq i < N$) ученик с номером i отдает свой лист ученику $P[i]$, где $P[0], P[1], \dots, P[N - 1]$ — это N **различных** целых чисел от 0 до $N - 1$ включительно. Перестановка P **фиксирована** на всю игру и одна и та же во всех раундах, но учителя и директор её не знают.

После завершения всех M раундов и выполнения последнего обмена согласно перестановке P в класс входит директор. Глядя только на листы, которые держат ученики, директор должен определить для каждого i ($0 \leq i < N$), номер раунда j на котором ученик i поднял руку, или сделать вывод, что ученик i никогда не поднимал руку.

Учителя не могут общаться с другими учителями или директором, за исключением массивов целых чисел, написанных ими на листах бумаги, которые ученики передают друг другу. Каждый учитель знает номер раунда, на котором он заходит в класс.

Они действуют независимо. Ваша задача — разработать и реализовать стратегию для учителей и директора школы, которая позволит правильно определить для каждого ученика, поднимал ли он руку, и когда это произошло, или сделать вывод, что ученик i никогда не поднимал руку. Ваш балл будет зависеть от максимальной длины массива,

написанного на листах бумаги — при уменьшении максимальной длины массива ваш балл может увеличиться или остаться таким же.

Детали реализации

Необходимо реализовать две функции: одну для учителей и одну для директора школы.

Для учителей вам следует реализовать следующую функцию:

```
std::vector<std::vector<int>> process_step(  
    int N, int M, int R, std::vector<int> T,  
    std::vector<std::vector<int>> A)
```

- N : количество учеников.
- M : количество раундов, а также количество учителей.
- R : текущий номер раунда (от 0 до $M - 1$).
- T : массив (возможно, пустой), содержащий индексы учеников, поднявших руку в этом раунде.
- A : массив длиной N , описывающий листы, где $A[i]$ ($0 \leq i < N$) — массив целых чисел на листе, который держал ученик i в начале раунда.
- Эта функция вызывается ровно M раз за игру в порядке $R = 0, 1, \dots, M - 1$.

Функция должна возвращать массив B длиной N , представляющий листы после внесения изменений учителем, в том же формате, что и A .

- Для каждого ученика i , поднявшего руку (то есть, число i встречается в массиве T), лист бумаги не должен быть изменен, поэтому $B[i] = A[i]$.
- Для каждого i такого, что $0 \leq i < N$, длина $B[i]$ не должна превышать 63, и каждое число в $B[i]$ должно находиться в диапазоне от 0 до 63 включительно.

Для директора школы следует реализовать следующую функцию:

```
std::vector<int> determine_steps(  
    int N, int M, std::vector<std::vector<int>> A)
```

- N, M : то же самое, что и выше.
- A : массив длины N , где $A[i]$ — массив целых чисел на листе бумаги, который находится у ученика i после всех M раундов.
- Эта функция вызывается ровно один раз за игру, после последнего вызова `process_step`.

Функция должна возвращать массив D длиной N . Для каждого i такого, что $0 \leq i < N$, должно выполняться следующее условие:

- $D[i] = j$, если ученик i поднял руку на раунде j , или
- $D[i] = -1$, если ученик i ни разу не поднял руку за всю игру.

Вашей программе запрещено хранить или передавать какую-либо информацию от одного вызова `process_step` к другому. Любое взаимодействие возможно только с использованием описанного в условии протокола, посредством аргументов функций и возвращаемых `process_step` значений через грейдер. Если ваша программа попытается нарушить это правило, ваш результат в CMS может быть неверным и может быть снижен после окончания тура. Обратите внимание, что тестирующая система может одновременно запускать несколько экземпляров вашей программы. Это означает, что вызовы `process_step` и `determine_steps` из одной и той же игры могут выполняться в разных экземплярах, а вызовы `process_step` и `determine_steps` из разных игр могут выполняться в одном и том же экземпляре. Каждый тест включает до 5 игр.

Ограничения

- $2 \leq N \leq 63$
- $1 \leq M \leq 63$
- Каждый ученик поднимает руку **не более одного раза** за всю игру. Другими словами, для каждого ученика i существует не более одного раунда j , на котором i поднимает руку.
- В каждом раунде как минимум один ученик **не** поднимает руку.

Подзадачи и оценка

Подзадача	Оценка	Дополнительные ограничения
1	4	$M = 1$
2	6	$N = 2$
3	9	$P[i] = i$ для всех $0 \leq i \leq N - 1$.
4	25	В каждом раунде поднимает руку не более одного ученика.
5	56	Без дополнительных ограничений.

Для каждого теста ваш балл равен 0 (в CMS отображается как `Output isn't correct`), если возвращаемое значение любого вызова функции `process_step` некорректно, или если возвращаемое значение любого вызова функции `determine_steps` неверно.

В противном случае пусть C будет максимальной длиной **любого** массива в **любом** B , возвращенном функцией `process_step`. Тогда оценка для теста в подзадаче с оценкой S вычисляется как $S \cdot X$, где X зависит от C согласно следующей таблице:

Условие	X
$C \leq 2$	1.00
$C = 3$	0.75
$C = 4$	0.55
$5 \leq C \leq 13$	$0.50 - 0.03 \cdot (C - 5)$
$14 \leq C \leq 63$	$0.19 \cdot \frac{64-C}{64-14} + 0.04$

Пример

Рассмотрим сценарий с $N = 4$ учениками, $M = 2$ учителями и перестановкой $P = [0, 3, 1, 2]$. Изначально все листы бумаги пусты.

Сначала грейдер делает вызов:

```
process_step(4, 2, 0, [1], [[], [], [], []])
```

Ученик 1 поднял руку, поэтому его лист не может быть изменен. Учитель 0 может решить написать $[10]$ на листе ученика 0, написать $[1, 63, 4]$ на листе ученика 3 и оставить лист ученика 2 пустым. Для этого функция должна вернуть $B = [[10], [], [], [1, 63, 4]]$.

После ухода учителя 0 ученики обмениваются листами в соответствии с P . После обмена у учеников находятся листы $A = [[10], [], [1, 63, 4], []]$.

Теперь грейдер делает вызов:

```
process_step(4, 2, 1, [0, 3], [[10], [], [1, 63, 4], []])
```

Ученики, поднявшие руки: 0 и 3, в их листы нельзя внести изменения. Учитель 1 может решить написать $[0, 40]$ на листе ученика 1 и написать $[1, 50]$ на листе ученика 2. Для этого функция должна вернуть $B = [[10], [0, 40], [1, 50], []]$.

После ухода учителя 1 ученики снова обмениваются листами в соответствии с P . После обмена листами $A = [[10], [1, 50], [], [0, 40]]$.

Наконец, грейдер делает вызов:

```
determine_steps(4, 2, [[10], [1, 50], [], [0, 40]])
```

Эта функция должна вернуть массив $D = [1, 0, -1, 1]$ поскольку ученики 0 и 3 подняли руку в раунде 1, ученик 1 поднял руку в раунде 0, а ученик 2 никогда не поднимал руку. В этом

примере $C = 3$.

Пример грейдера

Формат ввода:

```
N M
Q[0] Q[1] ... Q[N-1]
P[0] P[1] ... P[N-1]
```

Здесь Q — массив длиной N , где $Q[i] = j$ если ученик i поднимает руку на раунде j , или $Q[i] = -1$ если ученик i ни разу не поднимает руку за всю игру.

Формат вывода:

После каждого вызова функции `process_step`, грейдер выводит содержимое листов. Пусть K — длина B , а $L[i]$ — длина $B[i]$ (для каждого $0 \leq i < K$). Грейдер выводит B в следующем формате, **за которым следует пустая строка**:

```
K
L[0] B[0][0] B[0][1] ... B[0][L[0]-1]
L[1] B[1][0] B[1][1] ... B[1][L[1]-1]
:
L[K-1] B[K-1][0] B[K-1][1] ... B[K-1][L[K-1]-1]
```

Затем грейдер перемещает листы в соответствии с перестановкой P . Если $K \neq N$, грейдер выводит сообщение об ошибке и завершает работу.

После вызова функции `determine_steps`, грейдер выдает следующий результат:

```
C H
D[0] D[1] ... D[H-1]
```

Здесь H — это длина массива D возвращаемого функцией `determine_steps`.