

## Задача А. Пирамида

Имя входного файла:	стандартный ввод
Имя выходного файла:	стандартный вывод
Ограничение по времени:	1 секунда
Ограничение по памяти:	64 мегабайта

Для строительства двумерной пирамиды используются прямоугольные блоки, каждый из которых характеризуется шириной и высотой. Можно поставить один блок на другой, только если ширина верхнего блока строго меньше ширины нижнего. Самым нижним в пирамиде может быть блок любой ширины. По заданному набору блоков определите, пирамиду какой наибольшей высоты можно построить из них.

### Формат входных данных

В первой строке входных данных задается число  $N$  — количество блоков ( $1 \leq N \leq 100\,000$ ). В следующие  $N$  строках задаются пары натуральных чисел  $w_i$  и  $h_i$  ( $1 \leq w_i, h_i \leq 10^9$ ) — ширина и высота блока соответственно.

### Формат выходных данных

Выведите одно целое число — максимальную высоту пирамиды.

### Примеры

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

### Замечание

В примере пирамида будет состоять из двух блоков: нижним блоком будет блок номер 3, а верхним — блок номер 2. Блок номер 1 нельзя использовать вместе с блоком номер 3.

### Система оценки

Решения, правильно работающие при  $1 \leq N, w_i, h_i \leq 100$  будут набирать не менее 50% баллов.

## Задача В. Бонусмэн

Имя входного файла:	стандартный ввод
Имя выходного файла:	стандартный вывод
Ограничение по времени:	15 секунд
Ограничение по памяти:	512 мегабайт

Поле для игры «Бонусмэн» представляет собой поле из  $N$  строк и  $M$  столбцов. Каждая клетка поля может быть стеной (обозначается буквой "W"), бонусом (обозначается буквой "B") или пустой клеткой (обозначается символом ".").

Игрок может разместить на пустой клетке ловушку для бонусов. Ловушка для бонусов собирает все бонусы, находящиеся в том же ряду или столбце до ближайшей стены или края поля. То есть, захватываются все бонусы, до которых можно дойти от ловушки двигаясь влево, вправо, вверх и вниз, до тех пор, пока при движении в выбранном направлении не окажется клетки со стеной или края поля.

Нумерация игрового поля начинается с левого верхнего угла, клетки нумеруются с единицы.

Определите, в какой пустой клетке нужно разместить ловушку, чтобы собрать как можно больше бонусов.

### Формат входных данных

В первой строке задаются числа  $N$  ( $1 \leq N \leq 1000$ ) и  $M$  ( $1 \leq M \leq 1000$ ).

В следующих  $N$  строках задано описание поля для игры. Каждая строка состоит из  $M$  символов "B", "W" и ".".

### Формат выходных данных

Выведите номер строки и столбца пустой клетки, размещение ловушки в которой позволит собрать наибольшее количество бонусов. В случае, если ответов несколько — выведите любой из них.

### Примеры

стандартный ввод	стандартный вывод
3 3 WBW B.B WBW	2 2
3 4 .... .... ....	1 1
1 10 .BBWBB.BBW	1 7

### Замечание

Тест 1: 3 3 WBW B.B WBW Ответ 1: 2 2 Тест 2: 3 4 .... .... .... Ответ 2: 1 1 Тест 3: 1 10 .BBWBB.BBW  
 Ответ 3: 1 7 Пояснение: В тесте 2 можно разместить ловушку в любой клетке поля: количество собранных бонусов всегда будет равно 0.

## Задача С. Filler

Имя входного файла:	стандартный ввод
Имя выходного файла:	стандартный вывод
Ограничение по времени:	2 секунды
Ограничение по памяти:	64 мегабайта

Игра "Filler" происходит на прямоугольном клетчатом поле, где каждая клетка раскрашена в один из 6 цветов. Игра начинается с области, находящейся в верхнем левом углу поля. На каждом ходу игрок может изменить цвет области, содержащей верхнюю левую клетку поля, в результате чего к ней присоединяются соседние области того же цвета. Соседними считаются области, у которых есть две соседние по стороне клетки. Областью называется множество клеток одного цвета, где из каждой клетки можно попасть в каждую, двигаясь через сторону клетки.

Цель игры состоит в том, чтобы захватить правую нижнюю клетку игрового поля за наименьшее количество перекрашиваний.

Определите количество и порядок перекрашиваний.

### Формат входных данных

Вводится два числа  $N$  ( $1 \leq N \leq 2000$ ) и  $M$  ( $1 \leq M \leq 2000$ ) — размеры поля.

В каждой из следующих  $N$  строк содержится по  $M$  чисел от 1 до 6 — цвета клеток.

### Формат выходных данных

В первой строке выведите число  $K$  — минимальное количество перекрашиваний, необходимое для достижения правой нижней клетки поля.

Во второй строке выведите  $K$  чисел от 1 до 6 — последовательность перекрашиваний, которые необходимо осуществить.

Если правильных последовательностей перекрашиваний несколько — выведите любую из них.

### Пример

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

### Замечание

Пример 1: 3 5 1 2 3 5 6 3 4 3 6 6 4 3 1 6 5 Ответ 1: 4 2 3 6 5

## Задача D. Распределение по специализациям

Имя входного файла:	стандартный ввод
Имя выходного файла:	стандартный вывод
Ограничение по времени:	4 секунды
Ограничение по памяти:	256 мегабайт

На ФКН ВШЭ после 2 курса студенты выбирают себе специализацию. Каждый студент составляет список приоритетов специализаций, на которые он хотел бы попасть. Кроме того, у каждого студента есть его позиция в рейтинге (зависит от среднего балла за время обучения, у некоторых студентов позиция может совпадать). На каждой специализации есть максимальное количество студентов, которые могут быть на неё приняты.

Помогите справедливо распределить студентов в соответствие с их пожеланиями. Необходимо чтобы для каждого студента выполнялось следующее утверждение: после распределения студента на специализацию на более приоритетных для него специализациях не должно быть ни одного студента, с позицией в рейтинге больше чем у него. При этом из всех разбиений нас интересует такое, где наибольшее число студентов попало на специализацию.

### Формат входных данных

В первой строке задается два числа:  $N$  и  $K$  ( $1 \leq N, K \leq 100000$ ) — количество студентов и специализаций соответственно.

В следующей строке задается  $K$  натуральных чисел  $S_i$  ( $1 \leq S_i \leq 100000$ ), где  $S_i$  — количество студентов, которые готова принять специализация номер  $i$ .

В следующих  $N$  строках содержится описание приоритетов для каждого из студентов. Описание состоит из числа  $R$  ( $1 \leq R \leq N$ ) — позиции студента в рейтинге, числа  $T$  ( $1 \leq T \leq N$ ) — количества желаемых специализаций и  $T$  чисел от 1 до  $K$  — номера желаемых специализаций в порядке убывания приоритета. Номера специализаций не повторяются. Гарантируется, что сумма  $T$  для всех студентов не превосходит  $10^6$ .

### Формат выходных данных

Для каждого из  $N$  студентов выведите номер специализации, на которую он будет распределен. В случае, если на всех желаемых специализациях все места заняты студентами с более высокой позицией в рейтинге, то для студента следует вывести число -1. Если правильных ответов несколько — выведите любой из них.

### Пример

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

### Замечание

Решения, верно работающие в случае, если количество студентов и специализацией не превосходит 100 будут получать не менее 50% баллов.

## Задача Е. Описание карты

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

В давние времена карт почти не было, поэтому о возможности проехать из одного города в другой можно было узнать только из описаний других путешественников.

Всего было известно о существовании  $N$  городов, занумерованных числами от 1 до  $N$ . Сначала про существовании путей о дорогах между городами не было известно ничего, затем стали приходить отчёты от путешественников о возможности или невозможности добраться от одного города до другого, а также запросы о возможности или невозможности добраться от одного города до другого. Все дороги двусторонние, то есть если от одного города возможно (или невозможно) добраться до второго, то и от второго возможно (или невозможно) добраться до первого.

Помогите ученым, исследующим дорожную сеть давних времен, ответить на запросы. Запросы следует обрабатывать в том порядке, как они задавались, используя только имеющуюся на момент запроса информацию. Дорожная сеть в процессе работы не менялась, отчёты путешественников не содержат противоречий.

### Формат входных данных

В первой строке входных данных задаётся два числа:  $N$  и  $K$  ( $1 \leq N, K \leq 100000$ ) — количество городов и запросов.

В каждой из следующих  $K$  строк записан запрос одного из трех типов:

- +  $i j$  — существует путь по дорогам из города  $i$  в город  $j$ .
- $i j$  — не существует пути по дорогам из города  $i$  в город  $j$ .
- ?  $i j$  — определить, существует ли путь по дорогам из города  $i$  в город  $j$ .

### Формат выходных данных

На каждый запрос о существовании пути по дорогам выведите одну из трёх строк:

- + — если путь существует
- — если пути не существует
- ? — если по имеющимся на момент запроса отчётам нельзя точно определить существует или не существует путь между городами

### Примеры

стандартный ввод	стандартный вывод
3 4 + 1 2 ? 1 3 + 3 2 ? 1 3	? +
4 5 + 1 2 + 3 4 - 1 4 ? 2 4 ? 1 3	- -

### Замечание

Решения, верно работающие при  $1 \leq N, K \leq 1000$  будут получать не менее 50% баллов.