

## Задача 1. Поло

Имя входного файла:	input.txt
Имя выходного файла:	output.txt
Ограничение по времени:	1 секунда
Ограничение по памяти:	256 мегабайт

Аня и Катя — владельцы двух полей для игры в поло — «Ascot» и «Kirtlington» соответственно. Организаторы турнира обратились к ним со следующей задачей — необходимо провести  $2 \cdot N$  игр между  $N$  командами на двух полях.

Каждая из команд имеет две формы — белую и красную. Поэтому организаторы выдвинули следующее необычное требование — каждая из команд должна выступить на каждом поле два раза — один раз в белой форме, один раз — в красной, при этом в каждой игре формы команд должны отличаться. Но вот незадача, турнир уже завтра, жеребьевка прошла, и каждая команда знает, с кем и в какой форме она будет играть.

Так как владельцы поло-клубов — люди очень занятые, они попросили вас помочь им. Решите, возможно ли распределить игры между двумя полями и, если возможно, то покажите как.

### Формат входных данных

В первую строку входного файла записано одно число  $N$  — количество команд принимающих участие в турнире ( $2 \leq N \leq 10^5$ ). Каждая команда имеет свой номер от 1 до  $N$ .

Следующие  $2 \cdot N$  строк содержат по два номера команды:  $W_i$  и  $R_i$ , эта пара чисел означает, что команда  $W_i$  будет играть в белой форме с командой  $R_i$  в красной форме,  $i$  — номер игры ( $1 \leq i \leq 2 \cdot N; W_i \neq R_i$ ), жеребьевка гарантирует, что каждая команда два раза играет в белой форме и два раза в красной.

### Формат выходных данных

Если разделить игры между двумя полями невозможно, то в выходной файл необходимо вывести слово NO, иначе нужно вывести YES, а в следующих  $N$  строках указать номера игр, которые должны пройти на поле Ани. Номера нужно выводить в порядке возрастания. Если существует несколько вариантов ответа, можно вывести любой.

### Система оценки

Подзадача	Баллы	Ограничения	Необходимые подзадачи
1	20	$2 \leq N \leq 10$	
2	40	$2 \leq N \leq 1000$	1
3	40	$2 \leq N \leq 10^5$	1, 2

### Пример

input.txt	output.txt
3	YES
1 2	1
3 2	4
1 3	6
3 1	
2 1	
2 3	

## Задача 2. Увольнение

Имя входного файла: `input.txt`  
Имя выходного файла: `output.txt`  
Ограничение по времени: 2 секунды  
Ограничение по памяти: 256 мегабайт

Компания «DVD Projekt BLUE» имеет  $N$  сотрудников. В компании иерархия устроена так, что у каждого сотрудника, кроме гендиректора, есть прямой начальник. «Вассал моего вассала не мой вассал» — это не про эту компанию. Все подчинённые сотрудника будут подчинёнными и для его начальника.

К сожалению, компания разорилась и вынуждена сократить штат, но ещё не выбрала, кого именно увольнять. На повестке дня  $Q$  вариантов.

Инвесторы требуют, чтобы если оставшийся сотрудник был подчинённым для другого сотрудника, то и после сокращения он должен оставаться его подчинённым.

Помогите компании построить новые модели иерархии после возможных увольнений, если в каждом из  $Q$  вариантов руководство решает оставить  $K_i$  некоторых определённых сотрудников, где  $i$  — номер варианта.

### Формат входных данных

В первой строке содержится два целых числа:  $N$  и  $Q$  — количество сотрудников перед увольнениями и количество запросов ( $1 \leq N \leq 10^5$ ,  $1 \leq Q \leq 10^4$ ).

Следующие  $N$  строк содержат по два целых числа: идентификатор сотрудника  $s_r$  и идентификатор его прямого начальника  $h_r$  ( $1 \leq r \leq N$ ). Все идентификаторы сотрудников различные.

Если  $h_r = 0$ , то  $s_r$  — гендиректор компании. Гарантируется, что у компании есть гендиректор, все остальные сотрудники — его подчинённые, никакой сотрудник не является своим подчинённым ( $1 \leq s_i \leq N$ ,  $0 \leq h_i \leq N$ ,  $s_i \neq h_i$ ).

Далее описываются  $Q$  вариантов увольнений. Описание каждого варианта состоит из двух строк. Первая строка содержит одно целое число  $K_i$  — количество сотрудников, которых оставляют в компании ( $1 \leq i \leq Q$ ,  $1 \leq K_i \leq 20$ ,  $K_i \leq N$ ). Во второй строке записаны  $K_i$  различных целых чисел — идентификаторы этих сотрудников.

Гарантируется, что гендиректор не будет уволен.

### Формат выходных данных

В выходной файл требуется вывести для каждого варианта описание перестроенной иерархии. Это описание должно содержать для каждого оставленного сотрудника в отдельной строке два числа — идентификатор этого сотрудника и идентификатор его прямого начальника. Для гендиректора идентификатор прямого начальника равен 0.

Выводить строки можно в любом порядке в пределах варианта.

### Система оценки

Подзадача	Баллы	Ограничения	Необходимые подзадачи
1	31	$1 \leq N \cdot Q \leq 10^7$	
2	24	$1 \leq N \leq 10^4$ , $1 \leq Q \leq 10^4$	1
3	45	$1 \leq N \leq 10^5$ , $1 \leq Q \leq 10^4$	1, 2

### Пример

input.txt	output.txt
5 1	3 5
1 2	5 0
3 5	
4 3	
5 0	
2 5	
2	
3 5	

## Задача 3. Поймай крота

Имя входного файла: `input.txt`  
Имя выходного файла: `output.txt`  
Ограничение по времени: 1 секунда  
Ограничение по памяти: 256 мегабайт

Сергей любит играть на своем ноутбуке в вариацию известной игры «Поймай крота».

Игра происходит на прямоугольном клеточном поле, в клетках которого на краткое время появляются изображения кротов. Игрок стартует в момент времени, равный 1, из верхней левой клетки с координатами  $(1, 1)$ . За одну единицу времени игрок может передвигаться в клетку со смежной стороной. Как только игрок попадает в клетку с кротом, он его ловит. В любой момент времени игрок может либо передвигаться в соседнюю клетку, либо оставаться на месте и ждать.

В распоряжение Сергея попало описание нескольких уровней этой увлекательной игры. По этим описаниям он понял, что одновременно на поле может находиться только один крот.

Вам необходимо написать программу, которая посчитает, какое максимальное количество кротов можно поймать за игру.

### Формат входных данных

В первой строке входного файла заданы размеры поля  $N$ ,  $M$  и количество появляющихся в процессе игры изображений кротов  $K$  ( $1 \leq N, M \leq 100, 1 \leq K \leq 10^5$ ).

В следующих  $K$  строках описываются клетки, в которых появляются изображения кротов. Каждая строка содержит четыре числа: координаты клетки  $(x, y)$ , время появления крота  $t_s$  и время, когда его изображение пропадает  $t_e$  ( $1 \leq x \leq N, 1 \leq y \leq M, 1 \leq t_s < t_e \leq T \leq 10^5$ ).

В момент появления изображения крота уже можно поймать, а в момент, когда изображение пропадает, игрок не может поймать крота. Информация об изображениях задана в порядке их появления на поле. Гарантируется, что одновременно на поле не может находиться более одного крота.

### Формат выходных данных

В выходной файл необходимо вывести максимальное количество кротов, которое можно поймать на заданном уровне игры.

### Система оценки

Подзадача	Баллы	Ограничения	Необходимые подзадачи
1	18	$1 \leq K \leq 23, 1 \leq T \leq 10^5$	
2	19	$1 \leq K \leq 1000, 1 \leq T \leq 1000$	
3	28	$1 \leq K \leq 1000, 1 \leq T \leq 10^5,$ $1 \leq N, M \leq 15$	1, 2
4	35	$1 \leq K \leq 10^5, 1 \leq T \leq 10^5$	1, 2, 3

## Примеры

input.txt	output.txt
3 4 5 1 3 2 5 3 1 5 6 3 2 7 9 1 4 10 13 1 3 13 15	4
5 5 1 5 5 8 9	0

## Замечание

В первом примере игрок может успеть поймать четырёх кротов, один из вариантов — поймать 1-го, 3-го, 4-го и 5-го кротов.

Во втором примере игрок не успевает дойти до клетки (5, 5) и крот исчезает.

## Задача 4. Лиза и книга

Имя входного файла:	input.txt
Имя выходного файла:	output.txt
Ограничение по времени:	2 секунды
Ограничение по памяти:	256 мегабайт

Девочка Лиза написала свою собственную книгу и теперь хочет её издать. Книга разбита на главы, и Лиза хочет, чтобы каждая новая глава начиналась с новой страницы, даже если предыдущая страница заполнена ещё не полностью.

Лиза знает, сколько строчек занимает каждая глава её книги, а также сколько строчек помещается на одной странице. Но текущая версия книги не финальная, Лиза хочет вносить в неё ещё некоторые изменения. Например, она может объединить две соседние главы в одну, или наоборот — разбить одну главу на две другие. Ещё она может дописывать новые строки в существующие главы или удалять старые. В любой момент времени она хочет иметь возможность получать ответы на такие вопросы как: «Сколько страниц занимает книга?» и «Какая глава находится на странице с номером  $x$ ?». Помогите Лизе отвечать на эти вопросы.

### Формат входных данных

В первой строке входного файла записаны два числа  $N$  и  $M$  — количество глав в изначальной версии книги и количество строк, которые помещаются на одной странице ( $1 \leq N \leq 10^5$ ,  $1 \leq M \leq 10^9$ ).

Во второй строке записаны  $N$  чисел:  $a_1, \dots, a_N$ , где  $a_i$  — количество строк, которые занимает  $i$ -я глава ( $1 \leq a_i \leq 10^9$ ).

В третьей строке записано число  $Q$  — количество запросов ( $1 \leq Q \leq 10^5$ ).

Далее следуют  $Q$  строк, в каждой строке записан один запрос.

За  $N'$  будем обозначать текущее количество глав в книге. Главы и страницы нумеруются с 1. Запросы могут иметь следующий вид:

- **size** — вывести текущий размер книги в страницах;
- **chapter**  $x$  — вывести номер главы, которая находится сейчас на странице с номером  $x$  ( $1 \leq x \leq S$ , где  $S$  — текущий размер книги в страницах);
- **union**  $x$  — объединить главы с номерами  $x$  и  $x+1$  в одну главу с номером  $x$  ( $1 \leq x < N'$ ), размер новой главы равен сумме размеров старых, нумерация следующих глав при этом смещается;
- **split**  $x$   $y$  — разбить главу с номером  $x$  на две главы с номерами  $x$  и  $x+1$ , где глава  $x$  будет занимать  $y$  строк, а глава  $x+1$  будет занимать  $l-y$  строк, где  $l$  — размер главы  $x$  до выполнения запроса ( $1 \leq x \leq N'$ ,  $1 \leq y < l$ ), нумерация следующих глав при этом смещается;
- **resize**  $x$   $y$  — изменить размер главы с номером  $x$ , после выполнения запроса глава  $x$  занимает  $y$  строк ( $1 \leq x \leq N'$ ,  $1 \leq y \leq 10^9$ ).

### Формат выходных данных

Для каждого запроса вида **size** и **chapter** в выходной файл нужно выводить в новой строке ответы на эти запросы.

## Система оценки

Баллы за каждую подзадачу начисляются только в случае, если все тесты для этой подзадачи и необходимых подзадач успешно пройдены.

Подзадача	Баллы	Ограничения	Необходимые подзадачи
1	11	$1 \leq N, Q \leq 10^3$	
2	24	$1 \leq N, Q \leq 5 \cdot 10^4$	1
3	9	Допустимые запросы: <code>size</code> , <code>resize</code>	
4	16	Допустимые запросы: <code>chapter</code>	
5	40	Без дополнительных ограничений	1, 2, 3, 4

## Пример

input.txt	output.txt
3 4	6
5 6 8	7
9	6
size	1
split 3 2	5
size	2
union 1	
size	
chapter 3	
resize 1 5	
size	
chapter 3	

## Замечание

Пояснения к примеру из условия.

1. Начальный вариант:

_____	_____	_____	_____	_____	_____
_____	_____	_____	_____	_____	_____
_____	_____	_____	_____	_____	_____
_____	_____	_____	_____	_____	_____

2. После `split 3 2`:

_____	_____	_____	_____	_____	_____	_____
_____	_____	_____	_____	_____	_____	_____
_____	_____	_____	_____	_____	_____	_____
_____	_____	_____	_____	_____	_____	_____

3. После `union 1`:

_____	_____	_____	_____	_____	_____
_____	_____	_____	_____	_____	_____
_____	_____	_____	_____	_____	_____
_____	_____	_____	_____	_____	_____

4. После `resize 1 5`:

_____	_____	_____	_____	_____
_____	_____	_____	_____	_____
_____	_____	_____	_____	_____
_____	_____	_____	_____	_____

## Задача 5. Иннокентий и шифрование

Имя входного файла: `input.txt`  
Имя выходного файла: `output.txt`  
Ограничение по времени: 15 секунд  
Ограничение по памяти: 512 мегабайт

Иннокентий решил шифровать свои сообщения! Для этого он хочет применять побитовое исключающее ИЛИ к сообщению и ключу. Сообщения для шифрования у него есть, а вот ключей — нет. Поэтому он придумал свой способ их генерировать.

Напомним, что результат применения операции исключающее ИЛИ к двум битам равен единице, если только один из этих битов равен единице, иначе результат равен нулю. Побитовое исключающее ИЛИ (`xor`) применяется к битам, стоящим на одинаковых позициях в двух числах.

Например,  $10 \text{ XOR } 6 = 12$ .

В C/C++/Java — используется оператор `^`, в Python — функция `xor`.

В начале процесса генерации ключей последовательность чисел пуста. Иннокентий может применять операции двух типов:

- добавить число в конец последовательности;
- посчитать ключ для подпоследовательности рядом стоящих чисел.

Ключ Иннокентий решил считать следующим образом: для подпоследовательности, которая начинается с числа с номером  $l$  и заканчивается числом с номером  $r$ , он берет пару чисел  $x$  и  $k$ , выписывает последовательность длины  $r - l + 1$ , применяя к каждому числу побитовое исключающее ИЛИ с числом  $x$ . Затем он суммирует  $k$  наибольших значений в новой последовательности. Полученный результат и есть сгенерированный ключ.

Так как Иннокентию нужно отправить очень много новых сообщений, он не может сам считать ключи, и ему нужна ваша помощь.

### Формат входных данных

В первой строке входного файла задано одно число  $Q$  — количество операций ( $2 \leq Q \leq 10^5$ ).

В следующих  $Q$  числах заданы операции двух типов:

- $+ a$  — операция добавления числа  $a$  ( $1 \leq a \leq 10^9$ );
- $? l r x k$  — запрос вычисления ключа ( $1 \leq l \leq r \leq 2 \cdot 10^5$ ,  $k \leq r - l + 1$ ,  $1 \leq x \leq 10^9$ ).

### Формат выходных данных

Для каждого запроса вычисления ключа выведите одно число в новой строке выходного файла — значение ключа.

### Система оценки

Подзадача	Баллы	Ограничения	Необходимые подзадачи
1	4	$2 \leq Q \leq 20$ , $1 \leq a, k, x \leq 127$	
2	25	$2 \leq Q \leq 5000$ , $1 \leq a, k, x \leq 10^9$	1
3	31	$2 \leq Q \leq 10^5$ , $1 \leq a, k, x \leq 10^5$	1
4	40	$2 \leq Q \leq 10^5$ , $1 \leq a, k, x \leq 10^9$	1, 2, 3

## Пример

input.txt	output.txt
5 + 1 + 5 + 9 + 13 ? 1 4 2 3	33

## Замечание

В примере после добавлений получается следующая последовательность [1, 5, 9, 13]. В единственном запросе вычисления ключа, применив побитовое исключающее ИЛИ числа 2 с каждым из чисел исходной последовательности, получаем последовательность [3, 7, 11, 15]. Наибольшие 3 числа — 15, 11 и 7, в сумме дают 33.

## Задача 6. Дорога в парке

Имя входного файла: `input.txt`  
Имя выходного файла: `output.txt`  
Ограничение по времени: 1 секунда  
Ограничение по памяти: 256 мегабайт

Администрация парка решила проложить новую дорогу. В парке находятся две рощи: берёзовая и сосновая. Чтобы посетителям было приятно ходить по дороге, было решено сделать её прямой и проложить между этими рощами. Для этого был создан план парка: дерево на плане представлено точкой, а дорога — бесконечной прямой полосой некоторой ширины. Администрация хочет, чтобы ширина дороги была максимальной и чтобы ни одно дерево не находилось на дороге, но дерево может находиться на краю дороги. Важно, чтобы все деревья одного вида находились по одну сторону от дороги, а все деревья другого — по другую. Помогите администрации парка с реализацией проекта и напишите программу, которая найдет искомую дорогу.

### Формат входных данных

В первой строке входного файла заданы два целых числа  $N$  и  $M$  — количество деревьев в первой и во второй роще соответственно ( $1 \leq N, M \leq 10^5$ ).

В следующих  $N$  строках записаны пары чисел  $x, y$  — координаты берёз на плане парка ( $-10^6 \leq x, y \leq 10^6$ ).

Далее в  $M$  строках записаны пары чисел  $x, y$  — координаты сосен на плане парка ( $-10^6 \leq x, y \leq 10^6$ ).

Гарантируется, что существует хотя бы одна дорога ненулевой ширины, разделяющая рощи.

### Формат выходных данных

В первую строку выходного файла необходимо вывести координаты первой точки на середине дороги, во вторую строку — координаты второй точки середины дороги, на расстоянии не менее 1 от первой. В третью строку требуется вывести расстояние от центра дороги до её края.

Ваш ответ будет считаться правильным, если его абсолютная или относительная погрешность не превосходит  $10^{-4}$ .

### Система оценки

Подзадача	Баллы	Ограничения	Необходимые подзадачи
1	23	$1 \leq N, M \leq 20$	
2	20	$1 \leq N, M \leq 1000$	1
3	22	$1 \leq N, M \leq 50000$	1, 2
4	35	$1 \leq N, M \leq 10^5$	1, 2, 3

### Пример

input.txt	output.txt
3 3	0.00 0.00
-1 0	0.00 2.00
-3 0	1.00
-2 1	
1 0	
3 0	
2 1	

## Замечание

На рисунке ниже точки красного цвета — берёзы, синего — сосны. Чёрные точки — точки на середине дороги. Сама дорога — зелёная полоса.

