

Problem B. Point Pairs

Input file: *standard input*
Output file: *standard output*
Time limit: 2 seconds
Memory limit: 256 mebibytes

На плоскости заданы $2N + 1$ точек. i -я точка имеет координаты (X_i, Y_i) . Две точки i и j могут быть объединены в пару, если $X_i = X_j$ или $Y_i = Y_j$.

Для каждой точки, определите, можно ли разбить $2N$ точек, полученных из исходного набора выкидыванием данной точки, на N непересекающихся пар? For each point, determine the following:

Input

Формат ввода:

```
 $N$   
 $X_1$   $Y_1$   
 $X_2$   $Y_2$   
:  
 $X_{2N+1}$   $Y_{2N+1}$ 
```

Ограничения:

- $1 \leq N \leq 100,000$
- $1 \leq X_i, Y_i \leq 2N + 1$
- Все точки попарно различны.
- Все значения во входном файле целые.

Output

Выведите $2N + 1$ строк. В i -й из них выведите “OK”, если все точки, кроме данной, могут быть разбиты на N непересекающихся пар в соответствии с условиями задачи, и “NG” в противном случае.

Examples

standard input	standard output
1	NG
1 1	OK
1 2	OK
2 1	
2	OK
1 1	NG
1 2	OK
2 2	NG
2 3	OK
3 3	
2	NG
1 1	NG
1 2	OK
3 3	NG
4 4	NG
4 5	

Problem D. Nice Set of Points

Input file: *standard input*
Output file: *standard output*
Time limit: 1 second
Memory limit: 256 mebibytes

Рассмотрим множество точек.

Вы можете напрямую двигаться между двумя точками, если их x -координаты совпадают или их y -координаты совпадают. Назовём набор точек **хорошим**, если для любых двух точек в наборе длина кратчайшего пути равна манхэттенскому расстоянию между ними.

Вам заданы N точек. i -я точка имеет координаты (x_i, y_i) . Вы можете добавить не более $10^4 - N$ точек. Ваша задача — превратить заданный набор в хороший.

Input

Формат ввода:

```
 $N$   
 $x_1 y_1$   
 $x_2 y_2$   
:  
 $x_N y_N$ 
```

Ограничения:

- $2 \leq N \leq 1000$
- $1 \leq x_i, y_i \leq 1000$
- Точки попарно различны.
- Гарантируется, что хотя бы одно решение, удовлетворяющее условию задачи, существует.
- Все числа на входе целые.

Output

Пусть M ($0 \leq M \leq 10000 - N$) — количество добавленных точек, а $(s_1, t_1), \dots, (s_M, t_M)$ — их координаты. После добавления этих M точек в набор его размер станет $N + M$. Все $N + M$ точек **обязаны быть попарно различными** и набор должен быть хорошим. Координаты точек должны быть целыми.

Ответ выведите в следующем формате:

```
 $M$   
 $s_1 t_1$   
 $s_2 t_2$   
:  
 $s_N t_M$ 
```

Если решений несколько, выведите любое из них.

Examples

standard input	standard output
2 1 1 2 2	1 1 2
4 1 1 2 2 3 4 4 3	4 1 2 3 2 3 3 4 4
7 2 4 3 2 4 6 5 1 6 5 7 3 8 7	15 3 6 8 5 2 2 7 5 2 5 6 6 3 1 5 6 6 2 6 1 7 1 7 2 2 3 6 7 2 6

Note

В первом примере, если будет добавлена точка $(1, 2)$, вы сможете перейти между $(1, 1)$ и $(2, 2)$ через $(1, 2)$.

Problem G. Rectangle-free Grid

Input file: *no input*
Output file: *standard output*
Time limit: 1 second
Memory limit: 256 mebibytes

Постройте квадрат $N \times N$, составленный из символов 'O' (заглавная латинская буква O) и точек ('.'), обладающий следующими свойствами:

- $2 \leq N \leq 150$
- Не менее, чем 1700 из составляющих квадрат символов — буквы 'O'.
- Для каждой четвёрки целых чисел i, j, k, l такой, что $1 \leq i < j \leq N$ and $1 \leq k < l \leq N$, как минимум одна из четырёх клеток (i, k) , (i, l) , (j, k) , (j, l) содержит точку ('.').

Input

В этой задаче вход не предусмотрен.

Output

Первая строка должна содержать целое число N — сторону квадрата. Каждая из последующих N строк должна содержать N символов ('O' или '.'), и эти N строк задают квадрат.

Example

no input	standard output
	5

	..000
	...0.
	..0..

Note

Приведённый в примере ответ соответствует всем требованиям, кроме второго (общее количество букв 'O').

Problem H. Cups and Beans

Input file: *standard input*
Output file: *standard output*
Time limit: 1 second
Memory limit: 256 mebibytes

На столе стоят N чашек, занумерованных последовательными целыми числами от 0 до $N - 1$. Для каждого $i (1 \leq i \leq N - 1)$ чашка i содержит A_i бобов; также для неё определена константа C_i .

Двое играют в следующую игру:

- На каждом ходу игрок выбирает боб из одной из чашек, кроме нулевой 0.
- Если он выбрал боб из чашки i , он должен переместить его в одну из чашек с номерами $i - C_i, \dots, i - 1$.
- Игроки делают ходы по очереди. Если игрок не может сделать ход, он проигрывает. The players take turns alternately. If a player can't choose a bean, he loses.

Какой игрок выигрывает при оптимальной игре обоих?

Input

Формат ввода:

N
 $C_1 A_1$
 $C_2 A_2$
 \vdots
 $C_{N-1} A_{N-1}$

Ограничения:

- $2 \leq N \leq 10^5$
- $1 \leq C_i \leq i$
- $0 \leq A_i \leq 10^9$
- Как минимум одно из чисел A_i не равно нулю.
- Все числа во вводе целые.

Output

Выведите имя победителя: "First", если выигрывает первый игрок, и "Second" в противном случае.

Examples

standard input	standard output
3 1 0 1 1	Second
7 1 1 2 0 1 0 2 0 4 1 3 0	First
7 1 1 2 0 1 9 2 10 4 3 3 5	Second

Note

Пояснения к первому примеру:

- На первом ходу первый игрок переносит боб из чашки 2 в чашку 1.
- На втором ходу второй игрок переносит боб из чашки 1 в чашку 0.
- На третьем ходу первый игрок не может выбрать боб и проигрывает.

Problem J. Travel in Sugar Country

Input file: *standard input*
Output file: *standard output*
Time limit: 1.5 seconds
Memory limit: 256 mebibytes

В Сахарном Королевстве N городов, занумерованных последовательными целыми числами от 1 до N . Для каждого $i < N$ города i и $i + 1$ соединены дорогой с двусторонним движением, длина которой равна D_i .

Таким образом, для каждой пары городов (a, b) ($a < b$) расстояние между a и b равно $D(a, b) = D_a + D_{a+1} + \dots + D_{b-1}$.

В каждом городе есть магазин сахара. Муравей собирается посетить K различных магазинов.

Муравей выбирает набор из K различных магазинов и порядок, в котором он их посещает. Например, если он захочет посетить магазины в городах S_1, \dots, S_K в указанном порядке, он должен пройти расстояние $D(S_1, S_2) + D(S_2, S_3) + \dots + D(S_{K-1}, S_K)$.

Сколько существует различных вариантов выбора, при котором пройденное муравьём расстояние делится на M ? Выведите остаток от деления ответа на $10^9 + 7$.

Input

Формат ввода:

N M K
 D_1
 D_2
 \vdots
 D_{N-1}

Ограничения:

- $2 \leq N \leq 100$
- $1 \leq M \leq 30$
- $2 \leq K \leq 10, K \leq N$
- $1 \leq D_i \leq M$
- Все значения во входе целые.

Output

Выведите остаток от деления ответа на $10^9 + 7$.

Examples

standard input	standard output
4 4 3 2 1 3	6
15 5 10 5 5 5 5 5 5 5 5 5 5 5 5 5 5	897286330

Note

В первом примере существует следующие 6 путей, удовлетворяющих условию задачи: $1 \rightarrow 3 \rightarrow 2$, $2 \rightarrow 3 \rightarrow 1$, $2 \rightarrow 1 \rightarrow 4$, $4 \rightarrow 1 \rightarrow 2$, $2 \rightarrow 3 \rightarrow 4$ и $4 \rightarrow 3 \rightarrow 2$.

Problem K. Campus

Input file: *standard input*
Output file: *standard output*
Time limit: 2 seconds
Memory limit: 256 mebibytes

В связи с расширением территории кампуса руководство вашего университета приняло решение снести стену, ранее ограждавшую кампус (а сейчас отделяющую старую территорию кампуса от новой). Для выполнения этой задачи было нанято p рабочих.

У вас есть план стены — последовательность точек на плоскости с целыми координатами. При этом сегменты стены соединяют пары соседних точек и проходят параллельно осям координат. Также вы знаете длину стены, разбираемой одним рабочим за один час и количество нанятых рабочих.

Вам поручено по имеющимся данным вычислить время, за которое рабочие разберут стену.

Input

Первая строка входа содержит три целых числа n , s и f . $1 \leq n \leq 1000$ — количество сегментов в стене, $0 < s < 100$ — время в часах, за которое один рабочий разбирает один метр стены, $1 \leq p \leq 1000$ — количество рабочих.

Каждая из последующих $n + 1$ строк содержит два целых числа x_i и y_i ($-10^4 \leq x_i, y_i \leq 10^4$) — координаты i -й точки в последовательности; i -й сегмент стены соединяет точки с координатами (x_i, y_i) и (x_{i+1}, y_{i+1}) , при этом гарантируется, что или $x_{i+1} = x_i$, или $y_{i+1} = y_i$. Также гарантируется, что стена не пересекает саму себя.

Output

Выведите количество времени в часах, которое потребуется p рабочим, чтобы разобрать стену, округлённое вверх до целого числа (то есть если работа займёт 3.1 часа, вывести надо 4, а не 3).

Example

standard input	standard output
3 2 1 -2 4 1 4 1 2 3 2	14

Problem L. Race

Input file: *standard input*
Output file: *standard output*
Time limit: 2 seconds
Memory limit: 256 mebibytes

В процессе управления гоночной командой возникает много различных оптимизационных задач. Одна из них — выставить как можно больше гоночных машин в предстоящей гонке, распределив их между пилотами Вашей команды.

Для каждого пилота известен список машин, принадлежащих вашей команде, на которых он готов участвовать в гонке.

Ваша задача — распределить пилотов по машинам так, чтобы на старт вышло наибольшее количество машин.

Input

Первая строка входа содержит два целых числа $0 \leq m, n \leq 200$. m — количество машин, принадлежащих вашей команде, n — количество сотрудничающих с командой пилотов. Каждая из последующих n строк задаёт список предпочтения одного пилота и имеет следующий формат: i -я строка задаёт i -го пилота и начинается целым числом m_i ($0 \leq m_i \leq m$) — количества автомобилей, на которых этот пилот готов участвовать в гонке. Далее в строке следуют m_i целых чисел в интервале от 1 до m — номера машин, на которых i -й пилот готов участвовать. Все номера машин для одного пилота попарно различны.

Output

Выведите одно целое число — наибольшее количество машин, которое ваша команда сможет выставить на старт.

Example

standard input	standard output
5 5	3
1 2	
2 3 2	
2 2 3	
1 3	
4 2 1 3 4	

Problem M. Tip

Input file: *standard input*
Output file: *standard output*
Time limit: 2 seconds
Memory limit: 256 mebibytes

Когда известный тренер Майкл посещает свой любимый ресторан “Укконен и дерево”, он придерживается следующих правил.

- Он платит только целое количество долларов;
- Он всегда оставляет на чай не менее 0.2 суммы чека (то есть при чеке в 40 долларов Майкл оставляет на чай не менее 8, итого платит не менее 48 долларов, при чеке в 14 долларов он платит не менее 17 долларов (так как 0.2 от 14 это 2.80, а Майкл платит только целое количество долларов, то он платит не менее 17)).
- Если десятичная запись суммы, которую Майкл должен заплатить (включая чаевые) не является палиндромом, Майкл добавляет некоторое целое количество долларов к чаевым так, чтобы десятичная запись общей суммы была палиндромом (а добавленная к чаевым сумма при этом была минимально возможной).

По сумме чека за обед определите, какую сумму в итоге Майкл заплатит, следуя своим правилам.

Input

Первая строка входа содержит одно целое число T — количество тестовых примеров ($1 \leq T \leq 10^4$).

Каждый тестовый пример задаётся в отдельной строке и представляет собой одно целое число m ($5 \leq m \leq 10^4$) — чек Майкла в ресторане (в долларах).

Output

Для каждого тестового примера в отдельной строке выведите одно целое число — сумму, которую Майкл в итоге заплатит.

Example

standard input	standard output
2	22
14	101
83	

Problem N. Integer Triangle

Input file: *стандартный ввод*
Output file: *стандартный вывод*
Time limit: 2 seconds
Memory limit: 256 mebibytes

Заданы два целых числа a и b . Найдите, сколько существует различных целых чисел c таких, что треугольник со сторонами a , b и c имеет ненулевую площадь.

Input

В первой строке входа задано одно целое число T ($1 \leq T \leq 100$) — количество тестовых примеров.

Каждый тестовый пример задан в отдельной строке и состоит из двух целых чисел a и b ($1 \leq a, b \leq 100$).

Output

Для каждого набора в отдельной строке выведите одно целое число — количество различных целочисленных значений третьей стороны треугольника.

Example

стандартный ввод	стандартный вывод
1 4 3	5

Problem O. Арифметика

Input file: *standard input*
Output file: *standard output*
Time limit: 2 seconds
Memory limit: 256 mebibytes

Для заданных целых A и B определить общее количество чётных чисел, среди расположенных между A и B , включая и их.

Input

В единственной строке входа через пробел заданы числа A и B ($-10^9 \leq A, B \leq 10^9$).

Output

Выведите одно число — ответ к задаче.

Example

	standard input	standard output
1	1	0
2	-4	4