Table of Contents

Let's Cheer Up Bob 2
Standard input 2
Standard output 2
Constraints and notes 2

Quipu Function 5
Standard input 5
In the first line there are three integers t $(1 \leq t \leq 5)$, a and b $(1 \leq a \leq b \leq 10^{12}$ and $b - a \leq 10^5)$, where t represents the number of students (queries). In each one of the n lines there is one prime integer d_i $(d_i < 10^{12})$.
Standard output 5
Constraints and notes 5

Let's Cheer Up Bob

Time limit: *1280 ms*
Memory limit: *136 MB*

Bob is the world's worst tic-tac-toe player. He has never won a game. You, being a good friend, are determined to help him win his first game. You will insist that Bob makes the first move in the game.

In addition, you have noticed that Bob has been selecting squares in a methodical way. He has a definite preference order in which he selects squares. He chooses as his next square, the open square that is highest on his preference list. You will devise a list of moves that ensures, even with a really bad strategy, that Bob will be assured of a win.

From Wikipedia: *Tic-tac-toe (also known as noughts and crosses or Xs and Os) is a paper-and-pencil game for two players, X and O, who take turns marking the spaces in a 3×3 grid. The player who succeeds in placing three of their marks in a horizontal, vertical, or diagonal row wins the game.*

Standard input

The input consists of nine lines with the preference list of Bob's moves. Each line contains a two integers giving the row and the column of the moves. The rows and columns are numbers between 11 and 33, inclusive.

Standard output

The output will consist of a list of turns that you will make to ensure that Bob wins in the fewest possible moves. If at any point, you could make more than one move that would lead to a win for Bob in the fewest moves, you should choose the move that is in the lowest numbered row. If there are multiple moves in the lowest numbered row, you should choose the one with the lowest numbered column.

Constraints and notes

- The nine squares in the input are distinct

Input	Output	Explanation
1 2 3 3 3 1 1 1 2 2 2 3 3 2 2 1 1 3	1 1 2 2 2 3	The grid below gives the tic-tac-toe board with numbers reflecting Bob's preferences (lower numbers mean higher preference.) 4 1 9 8 5 6 3 7 2 Bob always moves first, and on his first move, he chooses the square with preference 1. If Bob is X's, the board now looks like: . X . . . . You choose the top left corner: O X . . . . Bob's highest remaining preference is 2, and the bottom corner is open, so after his second move, the board looks like: O X . . . . . . X You now take the middle: O X . . O . . . X

Input	Output	Explanation

Bob's 3rd preference is still available, so he chooses the lower left corner:

O	X	
	O	
X		X

You then move:

O	X	
	O	O
X		X

Bob's most preferred remaining square is the bottom middle one, so he moves and wins:

O	X	
	O	O
X	X	X

Whew! It is hard work making sure Bob wins, but that's just the kind of friend you are.

Quipu Function

Time limit: *1280 ms*
Memory limit: *18 MB*

Today is your first math class in Peru. Probably you know that Peruvian teachers include words in Quechua language in their classes.

In this context, your teacher defines a curious function $q(n,d)$ for a pair of positive integers n and d. This function is called *Quipu Function* and it's defined as the number of divisors of n which are not divisible by d.

The math class about *Quipu function* was amazing. After class, some students try to challenge the teacher. They ask questions like the following: "Given the positive integers a, b and d (where d is a prime number), can you tell me the result of next expression:

$\sum_{i=a}^{b} q(i,d)$ where q is the *Quipu Function*".

Unfortunately, the teacher got tired but he wants to answer all the questions by his students. Since he knows that you are the best programmer in class, he wants to know if you can help him by creating an application to compute the results for the student questions.

Standard input

In the first line there are three integers t $(1 \leq t \leq 5)$, a and b $(1 \leq a \leq b \leq 10^{12}$ and $b - a \leq 10^5)$, where t represents the number of students (queries). In each one of the next t lines there is one prime integer d_i $(d_i < 10^{12})$.

Standard output

For each query you need to write the result of the target expression given aa, bb and d_{i}d_i.

Constraints and notes

- $1 \leq t \leq 5$
- $1 \leq a \leq b \leq 10^{12}$
- $b - a \leq 10^5$
- $2 \leq d_i \leq 10^{12}$, d_i is prime

Input	Output	Explanation
2 3 6 3 5	8 10	In the sample input $t = 2$, so you have to answer to two questions where $a = 3$ and $b = 6$. In the first question, $d = 3$ so we need to sum the following: • $q(3, 3) = 1$: the divisors are $\{1, 3\}$, but 3 is not counted because $d \mid 3$ • $q(4, 3) = 3$: the divisors are $\{1, 2, 4\}$ • $q(5, 3) = 2$: the divisors are $\{1, 5\}$ • $q(6, 3) = 2$: the divisors are $\{1, 2, 3, 6\}$, but 3 and 6 are not counted because $d \mid 3$ and $d \mid 6$. So the sum $1 + 3 + 2 + 2 = 8$. In the second question, $d = 5$ so we need to sum the following: • $q(3, 5) = 2$: the divisors are $\{1, 3\}$ • $q(4, 5) = 3$: the divisors are $\{1, 2, 4\}$ • $q(5, 5) = 1$: the divisors are $\{1, 5\}$, but 5 is not counted because $d \mid 5$. • $q(6, 5) = 4$: the divisors are $\{1, 2, 3, 6\}$ So the sum $2 + 3 + 1 + 4 = 10$.

Note about this notation $x \mid y$: the $\mid$ sign means that x divides y, or that $\frac{y}{x}$ is an integer.

Vangelis the Batbear and the Bubbles challenge

Time limit: *1280 ms*
Memory limit: *264 MB*

Good evening master Wayne.

Joker and his gang attended Black Hat USA 2017 where they learned of a new way on how to damage our city! Specifically, tomorrow night they will try to damage the water pumps of Gotham using bubbles!

The bubbles cause corrosion to the pumps and in a few hours they will damage them with catastrophic results! To dash Joker's plan, besides stopping him, you need to make sure that the city network does not contain loops.

If Joker manages to inject bubbles to the network and they enter a loop, they will still cause damage to that area even though you would have already arrested Joker and his gang.

Given the map of the water distribution system, you need to make sure that the map does not contain loops.

Standard input

On the first line there will be an integer t, the number of test cases to follow.

For each test case, there will be 2 input lines:

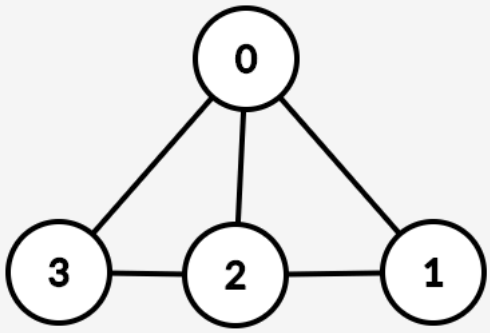
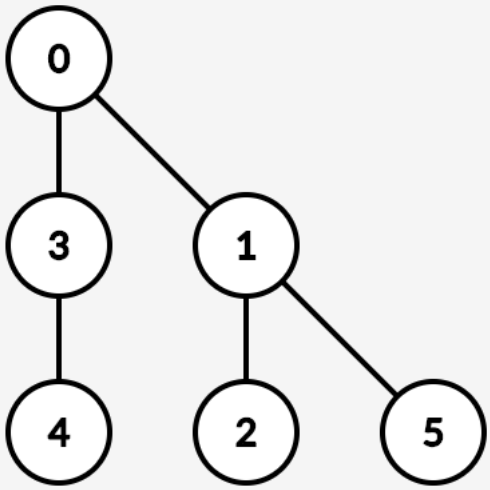
- On the first line of the test case, there will be 2 integers n and m, where n is the number of vertices and m is the number of edges.
- On the second line, there will be $m$$m$ pairs of integers separated by a space character. Each pair shows a two way connection between vertex a and vertex b.

Standard output

For each test case you will have to write one line that contains an integer, in the case where there is a loop you will write the number 1 or else you will write the number 0.

Constraints and notes

- $1 \leq t \leq 1\,000$
- $1 \leq n \leq 1\,000$
- $1 \leq m \leq 10\,000$
- $0 \leq a, b \leq n - 1$
- There can be multiple edges or self-loops. In this case we consider the graph to contain a loop.

Input	Output	Explanation
2 4 5 0 1 0 2 1 2 2 3 0 3 6 5 0 1 0 3 1 2 1 5 3 4	1 0	Test case #1  Test case #2 

Preparing for Xtreme 12.0

Time limit: *1280 ms*
Memory limit: *136 MB*

This is Alice's first programming contest. Since she has had such a great time, she has decided to prepare for Xtreme 12.0 in a methodical way. She asked her teammate, who has competed in previous Xtreme contests, for a list of up to 20 of the most important topics to study. With this list in hand, she borrowed a large number of books from the library, which collectively contain all of these topics. She is determined to read each of the books, cover to cover, but she does not have enough time to read them all.

Please help her plan for next year, by finding the minimum amount of time she will need to read a subset of these books, so that all of the important topics are covered.

Standard input

The input contains b $(1 \leq b \leq 100)$ lines which describe each book. These lines have the following form:

- $[\text{time}]$ $[\text{topic}_1]$ $[\text{topic}_2]$ ... $[\text{topic}_n]$

Where

- $[\text{time}]$ is the time, in minutes, required to read the book. This will be a non-negative integer less than 10^6.
- $[\text{topic}_i]$ is the i^{th} topic covered in the book.

Standard output

On standard output, print on a line by itself, the minimum time, in minutes, required for Alice to read books that cover all of the topics listed.

Constraints and notes

Constraints and notes

- $1 \leq b \leq 100$
- Each book will have up to 20 topics.
- The time requried to read a book is a non-negative integer less than 10^6
- There will be at most 20 total topics
- Each topic will be comprised of a string containing letters, numbers, and the underscore (but no whitespace).
- Each of the topics on the list will be covered by at least one book.

Input	Output	Explanation
300 Backtracking Dynamic_Programming Greedy 125 Dynamic_Programming 35 Backtracking 85 Greedy 120 Backtracking Dynamic_Programming 80 Greedy Backtracking	200	If Alice reads the last two books, she will cover all three of the topics in only 200 minutes. There is no way for her to cover these topics more quickly.

Sub Array Sum Problem

Time limit: *2480 ms*
Memory limit: *520 MB*

Given an 1D array the sub array sum problem asks for the sum of some subarrays (also called intervals). In 2D the same problem can be modelled as a 2D matrix and sub matrix sums, in 3D we have cubes. We can generalize this problem for every dimension.

Given a dd-dimensional array aa of integers answer queries of getting the sum of dd-dimensional subarray.

Standard input

The first line contains one integer d ($1 \leq d \leq 5$) which represents the number of dimensions of the array.

The second line contains d positive integers which represent a vector dim, each integer denoting the length of a dimension. The total size of the array is less than or equal to 10^5, formally $1 \leq \prod_{i=1}^{d} dim_i \leq 10^5$.

The third line contains $\prod_{i=1}^{d} dim_i$ integers, the values of the array coming in lexicographical order of their indices.

The fourth line contains one integer q ($1 \leq q \leq 10^5$) the number of queries.

Each query consists of two lines:

- First line of the query is a vector l of positive integers.
- Second line of the query is a vector r of positive integers.

Where:

- l is a d-tuple representing the coordinates of the lowest lexicographical index of the d-hyper rectangle.
- r is a d-tuple representing the coordinates of the highest lexicographical index of the d-hyper rectangle.
- Both vectors l and r are 1-based coordinates and $1 \leq l_i \leq r_i \leq dim_i$, $\forall i \mid 1 \leq i \leq d$

Standard output

For every query print the answer in a line , it guarantees the answer fit in a 32bit-signed integer.

Constraints and notes

- $1 \leq d \leq 5$
- $1 \leq \prod_{i=1}^{d} dim_i \leq 10^5$
- The elements of a fit in 32-bit signed integers
- $1 \leq q \leq 10^5$

Input	Output	Explanation
1 5 1 2 3 4 5 3 1 1 1 5 3 5	1 15 12	In the first test case we have an 1D array. For the first query we have 1. For the second query we have 1+2+3+4+5=15. For the third query we have 3+4+5=12.
2 3 3 1 2 3 4 5 6 7 8 9 2 1 1 1 1 1 1 2 2	1 12	In the second test case we have a 2D array. For the first query we have 1. For the second query we have 1+2+4+5=12.

Blackgate Penitentiary

Time limit: *1280 ms*
Memory limit: *264 MB*

Vangelis the Batbear trapped all the members of Joker's Streetgang in a basement.

Your job as a police officer is to transport all gang members to Blackgate Penitentiary.

To facilitate the transport, you should form a row such that the heights of the gang members are in **non-decreasing** order. For each gang member you should find the minimum and the maximum position where they can be in a valid sorted row and produce a roster with this information.

Standard input

Input will start with a line that contains only one integer n, the number of crew members that were arrested. On each of the following n lines there will be a single word s and an integer h separated by a space character, where s is the name and h is the height of the crew member.

Standard output

On the output, there will be gg lines. Each line will contain in alphabetical order and space separated the names of the crew members that have the same height, followed by the minimum and the maximum position where any member of the specific group can be placed. The groups should be printed in increasing order of their members' heights.

Constraints and notes

- $1 \leq n \leq 1\,000$
- $1 \leq \text{length}(s_i) \leq 10$
- $120 \leq h_i \leq 250$
- Names are only composed of characters of the Latin alphabet.

Input	Output
6 TheJoker 180 HarleyQuin 160 MrHammer 220 Boody 220 Muggs 180 Paulie 180	HarleyQuin 1 1 Muggs Paulie TheJoker 2 4 Boody MrHammer 5 6
10 a 200 aa 200 ab 200 aba 200 aaa 200 b 200 A 200 Aa 200 AB 200 B 200	A AB Aa B a aa aaa ab aba b 1 10

King Capture

Time limit: *2080 ms*
Memory limit: *136 MB*

TemplateInteractive

In this problem you should play a game on an undirected graph with n ($1 \le n \le 100$) vertices and m ($n - 1 \le m \le \binom{n}{2}$) edges. Initially you place a *white* chess king in one of the vertices, then your opponent places a *black* king in another vertex.

Following the initial placement, you and your opponent start performing alternating moves, you being the first to move.

At each turn, the player to move can do one of the following:

- Move her own king (you play with the white king, your opponent with the black one) in one of the vertices adjacent to the current vertex of the king.
- Do nothing

Game goal

You win if at any moment the two kings are in the same vertex.

Your opponent's moves will be performed by a special program called the *interactor*. It is guaranteed that you have a winning strategy on all the given tests. Even more, you want to win the game in as few moves as possible. The interactor will try to maximise the number of moves it takes you to win.

Interaction

First you should read from the standard input two integers n and m.

Each of the next m lines contains 2 integers, representing two vertices that share an edge.

Then you have to print the vertex you initially place the white king. After that you should read the initial position of the black king.

Next, start performing your moves. At each step print the index of the node where you want to move the king, or the index of the current node if you do nothing. Then you should read the index of the node where the interactor moves the black king.

Your program should exit after performing a move that brings the two kings on the same vertex.

Warning: Don't forget to flush after every output operation!

Constraints and notes

- This task is **adaptive**.
- $2 \leq n \leq 100$
- $n - 1 \leq m \leq \binom{n}{2}$
- The graph is connected, and it doesn't contain multiple edges or self-loops.

Interaction

```
7 11
1 3
1 5
1 6
2 3
2 5
2 6
2 7
3 6
3 7
4 5
5 6
```

```
2
```

```
1
```

```
6
```

```
3
```

```
3
```

```
7 11
1 5
1 6
1 7
2 3
2 4
2 5
3 4
3 5
3 6
4 5
5 6
```

Interaction

	1
2	
	5
3	
	3

Gotta Catch 'Em All

Time limit: *380 ms*
Memory limit: *136 MB*

Pokémon trainer Blue is starting his journey in the world of Pokémon and dreams of becoming a Pokémon master. Blue will be travelling to Viridian city for his first gym battle with his Pokémon Pikachu, but, in order to do so he has to pass through the Viridian forest. Blue's rival, Red is determined to stop Blue from becoming a Pokémon master. Hence, Red has laid traps in the Viridian forest with enemy Pokémon. These Pokémon will prevent Blue from proceeding further into the forest unless his Pikachu battles and defeats them. For each battle, Pikachu loses a portion of its health. Lucky for Blue, there are some unused health potions available in the forest. These potions can restore Pikachu's health. Can Pikachu fight all the enemy Pokémon? Will Blue reach the Viridian city? Or will Red be successful in his attempt to prevent Blue from becoming Pokémon master?

The Viridian forest can be represented as an $r \times c (2 \leq r, c \leq 200)$ grid. Blue starts from location $(1, 1)$ (which is on the top left part of the map) and needs to exit the Viridian forest from location (r, c) in order to go to Viridian city. At any cell (i, j), Blue can only move down $(i + 1, j)$ or right $(i, j + 1)$ and he cannot move outside the grid. In any cell (i, j) in the grid, there can either be a health potion or an enemy Pokémon which battles with Pikachu. Potions will be represented using positive integer and will increase Pikachu's health the value in the cell. Enemy Pokémon will be represented using negative integers and will reduce Pikachu's health by the value in the cell. If Pikachu's health drops to 0 or less, Pikachu will no longer be able to fight and Blue will have to go back to his home. Then Red will be successful in his attempt and Blue will never complete his dream of becoming a Pokémon master.

Blue does not want to lose, but, is unable to plan his journey through the Viridian forest. So, he needs your help to plan the journey for him so that Pikachu trains until he increases his health capacity to the minimum health needed to defeat all the Pokémon in their journey to Viridian city.

Standard input

On the first line of the input there will be two integers r and c which give the size of the Viridian forest.

On the following r lines there will be a c integers, each one representing the value of each cell in the forest (i, j) starting from position $(1, 1)$. Cells containing negative values contain enemy Pokémon and cells with positive values contain potions.

Cells $(1, 1)$ and (r, c) are always 0.

Standard output

Output a single integer, the minimum health that Pikachu needs for the journey.

Constraints and notes

- $2 \leq r, c \leq 200$
- The elements of the grid are integers in the interval $[-1\,000, 1\,000]$

Input	Output	Explanation
2 2 0 -2 2 0	1	Blue can take the path from $(1, 1)$ -> $(2, 1)$ -> $(2, 2)$. Pikachu needs to have health of 1 since there are no battles to be fought.
3 3 0 1 2 -2 -9 -4 -8 5 0	2	Blue can take the path from $(1, 1)$ -> $(1, 2)$ -> $(1, 3)$ -> $(2, 3)$ -> $(3, 3)$. Pikachu needs to have health of 2 since there is 1 battle to be fought and there are 2 potions in the path.

Elementary

Time limit: *1280 ms*
Memory limit: *136 MB*

Write a program that converts a word in the English dictionary (a string of lower case letters) into a number. Below are some examples of input-output pairs and there is a server that can give you more outputs for given inputs.

Your task is to find out what the functions does and program it.

Standard input

On the first line there will be an integer n*n*, the number of lines that will follow.

Lines 2, ..., n + 1will contain one word on each one.

Standard output

Print a list of *n* numbers, one per line.

For the given tests, each number will be smaller or equal to 10^6.

Input	Output

```
15
ieeextreme
one
brainteaser
snow
unicorn
laparoscopy
overcautiousness
cosmos
conclusion
wisconsin
binationalism
barbascos
cacophonic
inosculate
pneumonoconiosis
```

```
0
1
2
3
4
5
6
7
8
9
10
11
12
13
48
```

```
6
monkey
frog
bison
chinook
puffin
rhinoceros
```

```
0
1
2
3
4
6
```

```
5
kale
asparagus
broccoli
spinach
chocolate
```

```
0
1
2
3
4
```

Sample Tester

Here's a helper so you can test the server:

- Modify the input (left), and hit compute.
- You are allowed to interrogate at most 100100 strings of length at most 100100, each.
- There is a limit of 1010 interrogations per minute.

Compute

Math Challenge

Time limit: *680 ms*
Memory limit: *72 MB*

Paola is a strict Math teacher.

She challenged her students to calculate $a^{\binom{b}{c}}$ where $\binom{b}{c} = \frac{b!}{c!*(b-c)!}$.

Are you up to the challenge? Can you calculate this number?

Standard input

The input begins with an integer t $(1 \leq t \leq 10)$, giving the number of test cases in the input.

Each of the following t lines represent a different test case and consist of three integers a, b, c ($1 \leq a, b, c \leq 10^6$ and $b \geq c$).

Standard output

Output t lines, for each case print the result modulo $10^9 + 7$.

Constraints and notes

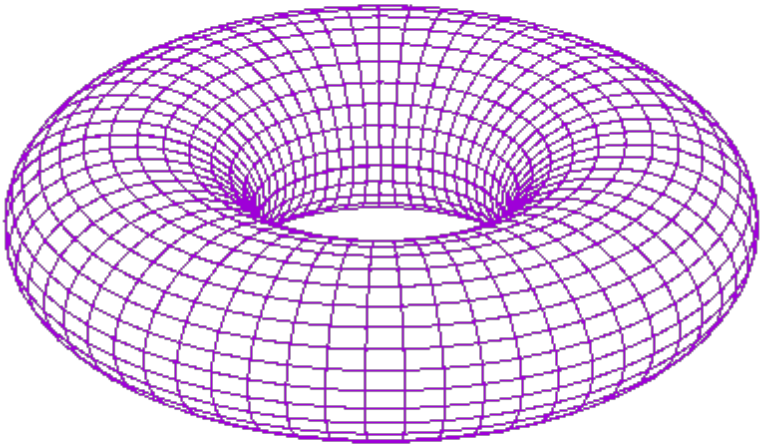
- $1 \leq t \leq 10$
- $1 \leq a, b, c \leq 10^6$
- $b \geq c$

Input	Output
3 1 5 5 2 10 2 3 9 4	1 371842544 665507626
3 2 5 3 3 8 4 1 1000000 500000	1024 662963356 1

Game of Life

Time limit: *1520 ms*
Memory limit: *136 MB*

You are asked to make an implementation of the game of life by John Horton Conway on a finite torus board and provide its output after cc iterations.



Standard input

On the first line of the input there will be three integers n, m ($1 \leq n, m \leq 25$) and c ($1 \leq c \leq 10^7$). n and m give the size of the board and c gives the number of iterations that you must simulate.

On the following n lines there will be m characters, either `*` or `-`, each one representing the value of each cell of the board. `*` represents a *populated* cell and `-` an *unpopulated* one.

Please note that the bottom neighbours of the last line are cells in the top line, and the left neighbours of the first column are the cells of the last column.

Standard output

On the output there should be n lines of m characters each, which represent the state of the board after c iterations.

Constraints and notes

- $1 \leq n, m \leq 25$
- $1 \leq c \leq 10^7$

Input	Output
4 6 3 ------ ------ ------ -***--	--*--- ------ --*--- --*---
5 6 1234 ------ --*--- ---*-- -***-- ------	-----* ---*-* ----** ------ ------

Input	Output
7 9 5 --------- --------- --------- --*****-- --------- --------- ---------	----*---- ---***--- --*-*-*-- -***-***- --*-*-*-- ---***--- ----*----
4 6 1 -----* --*--- *----- -***-*	**-**- ------ *--*-- -**-**

The Fast and the Curious

Time limit: *2480 ms*
Memory limit: *264 MB*

Jason Bourne is driving his car through the countryside, desperately trying to escape the black ops group «Operation Treadstone», that is on his tail. He is navigating through a map of the area and he can spot n ($1 \leq n \leq 10^5$) cities in the vicinity, numbered from 1 to n, connected by m two-way roads between them. There is at most one road between any two cities.

Jason must avoid «looping routes» at all costs because if he ends up in one, he can get trapped and caught. A «looping route» is a route of at least three cities, where each city is connected to the previous city with a road, the last city is also connected to the first city with a road and no city appears twice in the route. Jason knows that the only way to escape Treadstone, is by avoiding all cities that form «looping routes». Help Jason escape by finding the remaining cities on the map.

Standard input

The first line of the input consists of two space-separated integers, n and m. n is the number of cities on the map and m is the roads that connect them.

The next m lines each contain two integers a_i and b_i, indicating that there is a road between city a_i and city b_i.

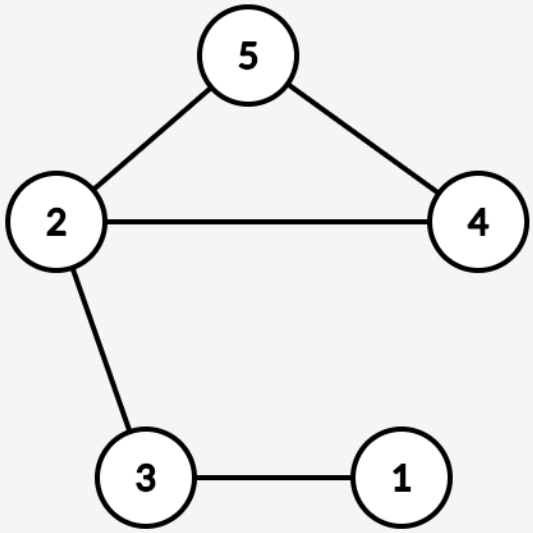
Standard output

Print the numbers of the cities which are not on any looping route, one per line, in ascending order. If no such city exists, do not print anything.

Constraints and notes

- $1 \leq n \leq 10^5$
- $0 \leq m \leq 3 * 10^5$
- $1 \leq a_i, b_i \leq n, a_i \neq b_i$
- There is at most one road between any two cities.

Input	Output	Explanation
5 5 4 5 5 2 2 4 1 3 3 2	1 3	In the map below, there are 5 cities. Cities 2, 4 and 5 form a looping route, but cities 1 and 3 are not on any looping routes.

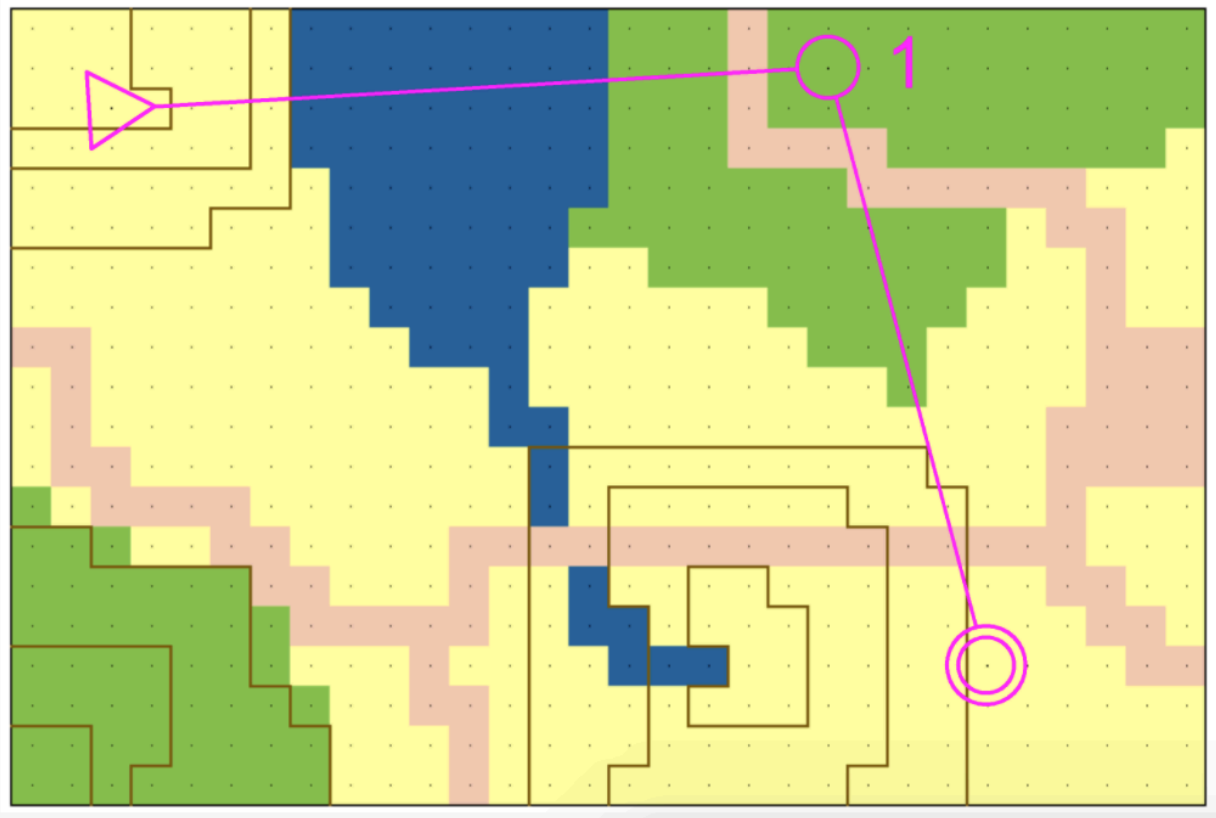


Orienteering

Time limit: *3680 ms*
Memory limit: *136 MB*

Orienteering is a wonderful outdoor sport in which participants are expected to reach a set of control points in the specified order. The path is not predetermined and each runner navigates by its own with the help of a detailed map and a compass. During the race, orienteers face multiple challenges, including: planning the best path from each control point to the following, navigating into the terrain to follow the desired path and to avoid getting lost, and running as fast as possible to beat other competitors.

In this challenge, you deal with path planning: given an orienteering map and a course, which is the best *route choice*? To answer this question consider a grid map representation: a uniform subdivision of the terrain into 10×10m squares called *tiles*. So, the course is a sequence of $p + 1$ tiles in the grid: the starting tile and p tiles in which control points are located.



Orienteering

Time limit: *3680 ms*
Memory limit: *136 MB*

Orienteering is a wonderful outdoor sport in which participants are expected to reach a set of control points in the specified order. The path is not predetermined and each runner navigates by its own with the help of a detailed map and a compass. During the race, orienteers face multiple challenges, including: planning the best path from each control point to the following, navigating into the terrain to follow the desired path and to avoid getting lost, and running as fast as possible to beat other competitors.

In this challenge, you deal with path planning: given an orienteering map and a course, which is the best *route choice*? To answer this question consider a grid map representation: a uniform subdivision of the terrain into 10×10m squares called *tiles*. So, the course is a sequence of $p+1$ tiles in the grid: the starting tile and p tiles in which control points are located.

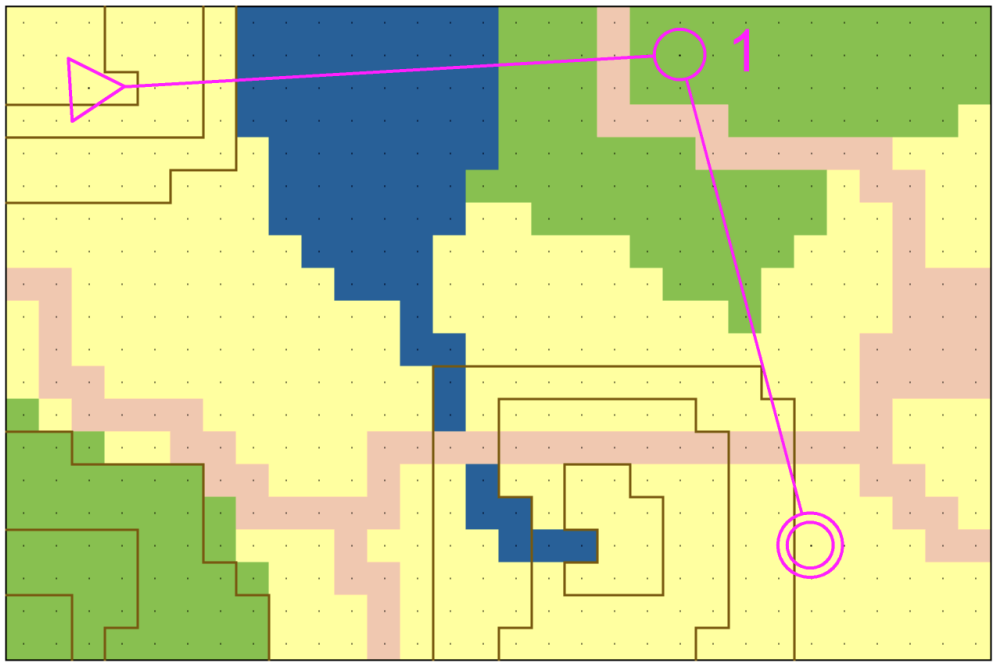


Illustration 1 *An orienteering map with a course. The course is represented in purple and is composed by three points: the start (indicated with a triangle), the first control point (single circle) and the final control point (double circle). In the map can be recognized some terrain features: a small river and a marsh (in blue), streets (in pink), grass fields (yellow) and woods (green). Moreover, in the map are represented also three hills with contours (brown lines).*

In this context, the orienteering map is encoded with two matrices, representing terrain characteristics. The first matrix r contains information about the type of terrain and in particular its runnability: for example, a grass field has a better runnability than a marsh. In this matrix, values are proportional to the expected crossing time of tiles, so smaller values indicate better runnability. The second matrix h is the digital elevation model (DEM) of the terrain: element values represent tile's altitudes.

3	3	3	3	3	3	3	9	9	9	9	9	9	9	9	5	5	5	2	5	5	5	5	5	5	5	5	5	5	5
3	3	3	3	3	3	3	9	9	9	9	9	9	9	9	5	5	5	2	5	5	5	5	5	5	5	5	5	5	5
3	3	3	3	3	3	3	9	9	9	9	9	9	9	9	5	5	5	2	5	5	5	5	5	5	5	5	5	5	5
3	3	3	3	3	3	3	9	9	9	9	9	9	9	9	5	5	5	2	2	2	2	5	5	5	5	5	5	5	3
3	3	3	3	3	3	3	3	9	9	9	9	9	9	9	5	5	5	5	5	5	2	2	2	2	2	2	3	3	3
3	3	3	3	3	3	3	3	9	9	9	9	9	9	5	5	5	5	5	5	5	5	5	5	5	3	2	2	3	3
3	3	3	3	3	3	3	3	9	9	9	9	9	9	3	3	5	5	5	5	5	5	5	5	5	3	3	2	3	3
3	3	3	3	3	3	3	3	3	9	9	9	9	3	3	3	3	3	3	5	5	5	5	5	3	3	3	2	3	3
2	2	3	3	3	3	3	3	3	3	9	9	9	3	3	3	3	3	3	3	5	5	5	3	3	3	3	2	2	2
3	2	3	3	3	3	3	3	3	3	3	3	9	3	3	3	3	3	3	3	3	3	5	3	3	3	3	2	2	2
3	2	3	3	3	3	3	3	3	3	3	3	9	9	3	3	3	3	3	3	3	3	3	3	3	3	2	2	2	2
3	2	2	3	3	3	3	3	3	3	3	3	3	9	3	3	3	3	3	3	3	3	3	3	3	3	2	2	2	2
5	3	2	2	2	2	3	3	3	3	3	3	3	9	3	3	3	3	3	3	3	3	3	3	3	3	2	3	3	3
5	5	5	3	3	2	2	3	3	3	3	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	3	3	3
5	5	5	5	5	5	2	2	3	3	3	2	3	3	9	3	3	3	3	3	3	3	3	3	3	3	2	2	3	3
5	5	5	5	5	5	5	2	2	2	2	2	3	3	9	9	3	3	3	3	3	3	3	3	3	3	3	2	2	3
5	5	5	5	5	5	5	3	3	3	2	3	3	3	3	9	9	9	3	3	3	3	3	3	3	3	3	3	2	2
5	5	5	5	5	5	5	5	3	3	2	2	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3
5	5	5	5	5	5	5	5	3	3	3	2	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3
5	5	5	5	5	5	5	5	3	3	3	2	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3

9	9	9	7	7	7	5	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4
9	9	9	7	7	7	5	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4
9	9	9	9	7	7	5	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4
7	7	7	7	7	7	5	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4
5	5	5	5	5	5	5	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4
5	5	5	5	5	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4
4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4
4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4
4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4
4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4
4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4
4	4	4	4	4	4	4	4	4	4	4	4	4	5	5	5	5	5	5	5	5	5	5	4	4	4	4	4	4	4
4	4	4	4	4	4	4	4	4	4	4	4	4	5	6	7	7	7	7	7	7	5	5	5	4	4	4	4	4	4
5	5	4	4	4	4	4	4	4	4	4	4	4	5	6	7	8	8	8	8	7	7	5	5	4	4	4	4	4	4
5	5	5	5	5	5	4	4	4	4	4	4	4	5	6	7	8	9	9	8	8	7	5	5	4	4	4	4	4	4
6	6	6	6	6	5	4	4	4	4	4	4	4	5	6	6	8	9	9	9	8	7	5	5	4	4	4	4	4	4
8	8	7	7	6	5	4	4	4	4	4	4	4	5	6	6	7	7	9	9	8	7	5	5	4	4	4	4	4	4
8	8	8	7	6	5	5	4	4	4	4	4	4	5	6	6	8	9	9	9	8	7	5	5	4	4	4	4	4	4
9	9	8	7	6	6	5	5	4	4	4	4	4	5	6	6	8	8	8	8	8	7	5	5	4	4	4	4	4	4
9	9	8	6	6	6	5	5	4	4	4	4	4	5	6	7	7	7	7	7	7	5	5	5	4	4	4	4	4	4

Illustration 2 *Map representation as two matrices: the runnability matrix (up) and the elevation matrix (down). Note that the runnability of the streets (2) is better than other types of terrain, for example runnability of marshes is 9.*

The travel time for moving from a tile to a neighboring tile depends on two factors: (i) the runnability of both tiles and (ii) the slope of the terrain, i.e., the difference in elevation of the tiles. In particular, the travel time follows the Tobler's hiking function, which is commonly used to determine the hiking speed taking into account the slope angle. In details, the travel time t from tile a to a neighbour tile b can be computed as follows: $t = \frac{r_a + r_b}{2} * e^{3.5|\frac{h_b - h_a}{10} + 0.05|}$, where r_x and h_x are the runnability and the elevation of the generic tile x, respectively.

To plan the path, consider tiles as 4-connected: a tile at the i^{th} row and j^{th} column of the matrix is a neighbour of at most 4 tiles: the tile at north $(i-1, j)$ if it exists, the tile at south $(i+1, j)$, the tile at east $(i, j+1)$ and the tile at west $(i, j-1)$. So, it is not possible in a path to have diagonal steps.

Task

Your task is to find the minimum time required to complete the course, which corresponds to the travel time of the best path. Remember that control points must be visited in the given sequence, and they can appear in the course more than once. In case a control point appears more than once in the course, it must be visited more than once.

Standard input

The first line contains two integers separated by a blank space: the number of rows n and the number of columns m of the map ($1 \leq n, m \leq 600$).

The second line contains a single integer p ($1 \leq p \leq 30$), which is the number of control points of the orienteering course. The last control point is the finish of the course.

The third line contains the starting point as two integers i, j separated by a blank space for representing the i^{th} row and the j^{th} column of the grid ($0 \leq i < n$ and $0 \leq j < m$).

The following p lines contain the positions of the control points, including the finish. The same representation as the one for the starting point is used. Note that the same position can appear more than once in the course.

The following n lines represent the runnability matrix and contain m rational numbers each. Numbers are separated by a blank space and use the *dot* as separator of the decimal part ($0 < r_{i,j} \leq 9,999.99$).

The following n lines represent the elevation matrix and have the same structure of the previous matrix; elevation is expressed in meters ($0 \leq h_{i,j} \leq 8,848.00$).

Standard output

Print a single integer representing the travel time for completing the course. Rounding by ceiling is required if the resulting travel time is not an integer.

Constraints and notes

- $1 \leq n, m \leq 600$
- $1 \leq p \leq 30$
- $0 < r_{i,j} \leq 9,999.99$
- $0 \leq h_{i,j} \leq 8,848.00$

Input	Output
1 3 2 0 0 0 2 0 0 5.0 6.0 5.0 100.0 101.5 103.0	38
20 30 2 2 2 1 20 16 24 3 3 3 3 3 3 3 9 9 9 9 9 9 9 9 5 5 5 2 5 5 5 5 5 5 5 5 5 5 3 3 3 3 3 3 3 9 9 9 9 9 9 9 9 5 5 5 2 5 5 5 5 5 5 5 5 5 5 3 3 3 3 3 3 3 9 9 9 9 9 9 9 9 5 5 5 2 5 5 5 5 5 5 5 5 5 5 3 3 3 3 3 3 3 9 9 9 9 9 9 9 9 5 5 5 2 2 2 2 5 5 5 5 5 5 5 3 3 3 3 3 3 3 3 3 9 9 9 9 9 9 9 5 5 5 5 5 5 2 2 2 2 2 2 3 3 3 3 3 3 3 3 3 3 3 9 9 9 9 9 9 5 5 5 5 5 5 5 5 5 5 5 3 2 2 3 3 3 3 3 3 3 3 3 3 9 9 9 9 9 9 3 3 5 5 5 5 5 5 5 5 5 5 3 3 2 3 3 3 3 3 3 3 3 3 3 3 9 9 9 9 3 3 3 3 3 3 5 5 5 5 5 3 3 3 2 3 3 2 2 3 3 3 3 3 3 3 3 9 9 9 3 3 3 3 3 3 3 5 5 5 3 3 3 3 2 2 2 3 2 3 3 3 3 3 3 3 3 3 3 9 3 3 3 3 3 3 3 3 3 5 3 3 3 3 2 2 2 3 2 3 3 3 3 3 3 3 3 3 3 9 9 3 3 3 3 3 3 3 3 3 3 3 3 2 2 2 2 3 2 2 3 3 3 3 3 3 3 3 3 3 9 3 3 3 3 3 3 3 3 3 3 3 3 2 2 2 2 5 3 2 2 2 2 3 3 3 3 3 3 3 9 3 3 3 3 3 3 3 3 3 3 3 3 2 3 3 3 5 5 5 3 3 2 2 3 3 3 3 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 3 3 3 5 5 5 5 5 5 2 2 3 3 3 2 3 3 9 3 3 3 3 3 3 3 3 3 3 3 2 2 3 3 5 5 5 5 5 5 5 2 2 2 2 2 3 3 9 9 3 3 3 3 3 3 3 3 3 3 3 2 2 3 5 5 5 5 5 5 5 3 3 3 2 3 3 3 3 9 9 9 3 3 3 3 3 3 3 3 3 3 2 2 5 5 5 5 5 5 5 5 3 3 2 2 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 5 5 5 5 5 5 5 5 3 3 3 2 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 5 5 5 5 5 5 5 5 3 3 3 2 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 9 9 9 7 7 7 5 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 9 9 9 7 7 7 5 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 9 9 9 9 7 7 5 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 7 7 7 7 7 7 5 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4	204

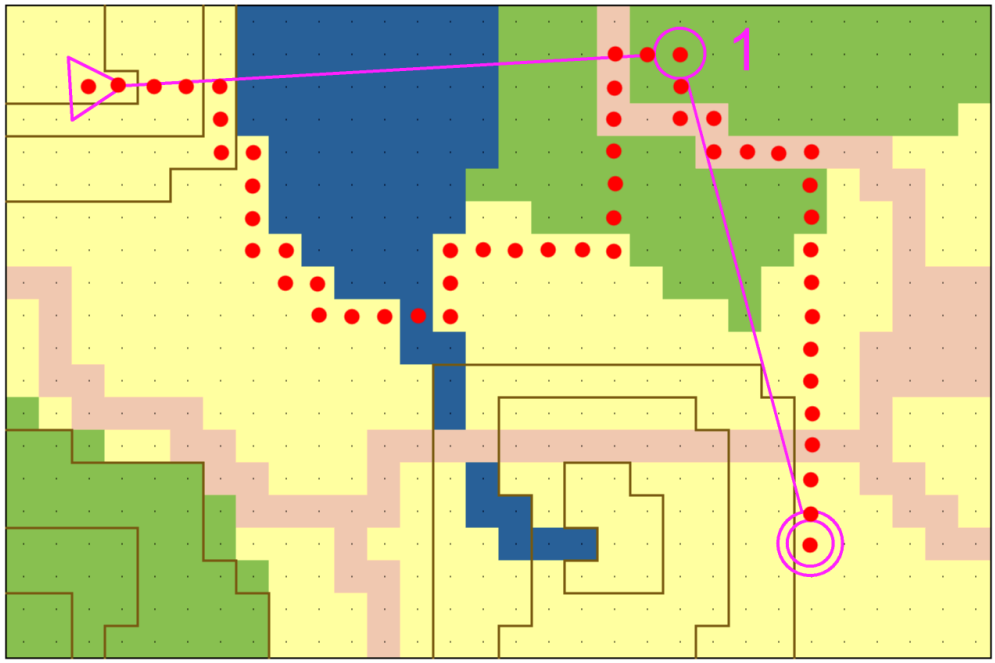
Input	Output

```
5 5 5 5 5 5 5 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4
5 5 5 5 5 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4
4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4
4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4
4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4
4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4
4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4
4 4 4 4 4 4 4 4 4 4 4 4 4 5 5 5 5 5 5 5 5 5 5 4 4 4 4 4 4 4
4 4 4 4 4 4 4 4 4 4 4 4 4 5 6 7 7 7 7 7 7 5 5 5 4 4 4 4 4 4
5 5 4 4 4 4 4 4 4 4 4 4 4 5 6 7 8 8 8 8 7 7 5 5 4 4 4 4 4 4
5 5 5 5 5 5 4 4 4 4 4 4 4 5 6 7 8 9 9 8 8 7 5 5 4 4 4 4 4 4
6 6 6 6 6 5 4 4 4 4 4 4 4 5 6 6 8 9 9 9 8 7 5 5 4 4 4 4 4 4
8 8 7 7 6 5 4 4 4 4 4 4 4 5 6 6 7 7 9 9 8 7 5 5 4 4 4 4 4 4
8 8 8 7 6 5 5 4 4 4 4 4 4 5 6 6 8 9 9 9 8 7 5 5 4 4 4 4 4 4
9 9 8 7 6 6 5 5 4 4 4 4 4 5 6 6 8 8 8 8 8 7 5 5 4 4 4 4 4 4
9 9 8 6 6 6 5 5 4 4 4 4 4 5 6 7 7 7 7 7 7 5 5 5 4 4 4 4 4 4
```

Explanation

In the first sample, there is a simple map with 1 row and 3 columns. The course starts at $(0, 0)$, has a control point at $(0, 2)$ and then returns to the starting position $(0, 0)$. Only one single path is possible in this case, so there is no route choice. Note that the path consist of four steps: $(0, 0) \to (0, 1)$, then $(0, 1) \to (0, 2)$, and then back $(0, 2) \to (0, 1)$, and $(0, 1) \to (0, 0)$. To compute the total travel time, we need to apply the given formula four times, one for each step, obtaining: 11.08, 11.08, 7.80, and 7.80. Note that the way back is faster because it is downhill. The expected output is given by the sum and the rounding by ceiling: $\text{ceil}(11.08 + 11.08 + 7.80 + 7.80) = \text{ceil}(37.76) = 38$.

In the second sample, input encodes illustrations of the problem statement. Note that, in the illustration of the matrices, values are depicted as integer, while in the input the dotted format for rational numbers is used. One of the best paths for completing the given course is represented in the following illustration:



The total travel time of the path can be computed step by step applying the given formula for neighbour tiles. It results to be equal to about $203.12203.12$. So, the desired solution is the ceiling value: 204204. Note that in this case other paths have an equivalent travel time of the given one, but it is the minimum.

Shuffle

Time limit: *1280 ms*
Memory limit: *136 MB*

A group of scientists wish to start an experiment in a village. It consists of swapping lodging of families for a week and then collecting data through a survey.

For the experiment to be a success, there are certain rules that should be followed, namely:

- No family should be sent to its own lodging,
- No family should be sent to a lodging it has already visited.

Under these conditions, the scientists would like to know if such a shuffle of lodgings is possible. If not, they are ready to send some families to a hotel and would like to know what is the minimum number of families that should be sent to the hotel in order to make the experiment become possible (note that when a family is sent to the hotel, its lodging can still be used for another family).

Standard input

The first line of the input contains the number n ($n \leq 200$), the number of families willing to participate in the experiment. We consider the families to be indexed from 0 to $n-1$.

The n remaining lines are such that the i^{th} line contains the list of indexes of families the family $(i-2)$ has already visited, separated with spaces.

Standard output

Print an integer corresponding to the minimum number of families to send to the hotel so that a shuffle of lodging becomes possible.

Constraints and notes

- $2 \leq n \leq 200$
- The families are indexed from 0 to $n-1$

Input	Output
5 1 2 3 2 3 4	0

Input	Output
3 4 0 4 0 1 0 1 2	
5 2 3 4 2 3 4 0 1 0 1 0 1	0
5 2 3 4 2 3 4 0 1 0 1 2 0 1 2	1

Octopuses with Watches

Time limit: *1280 ms*
Memory limit: *13 MB*

IBM puzzlemaster asked, on March 2017 a challenge about [eight octopuses where each one of them has eight watches](#).

Your challenge is to write a program for a generalization of the problem – to find the maximal number of watches that can be adjusted to either 33, 66, 99 or 1212 according to the rules of the game.

Using a sequence of operations of two types:

1. Add an hour to all the watches of a single octopus.
2. Add an hour to a specific watch for all the octopuses.

and given the hour setting of $n \times m$ watches (m watches for each of the n octopuses) compute the maximal number of watches that can be adjusted to either 3, 6, 9 or 12.

Standard input

On the first line of the input there will be two integers n and m ($0 < n \leq m < 10$).

On the following n lines there will be m integers in the interval $[1, 12]$, each one representing the configuration of one of the watches.

Standard output

Output a single integer, the maximal number of watches that can be adjusted to either 3, 6, 9 or 12.

Constraints and notes

- $0 < n \leq m < 10$

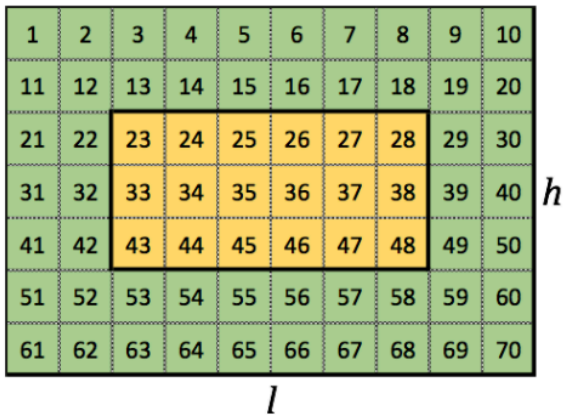
Input	Output
8 8 1 2 3 4 5 6 7 8 2 4 6 8 10 12 2 4 3 6 9 12 3 6 9 12 4 8 12 4 8 12 4 8	43

Input	Output
5 10 3 8 1 6 11 4 6 12 6 12 6 12 6 12 7 2 9 4 11 6 1 8 8 4 12 8 4 12 8 4	
3 4 3 1 4 1 5 9 2 6 5 3 5 8	10

RecXor

Time limit: *680 ms*
Memory limit: *264 MB*

A rectangle of dimensions $l \times h$ $(1 \leq l, h \leq 10^6)$ is filled with numbers sequentially starting from number n $(1 \leq n \leq 10^9)$ till the end of the rectangle. There is a second rectangle that fits inside the first rectangle and is defined by the end points of either of the diagonals, say d_1 and d_2 $(n \leq d1 \leq d2 < n + l * h)$ which denote the numbers the diagonal starts from and ends at respectively. Your task is to find the `xor` of all the numbers that are not common to both of the rectangles. (Images below for better understanding.)



Outer rectangle: 10 (*l*) x 7 *(h)*
Inner rectangle: 23*(d1)*, 48 *(d2)*

Outer rectangle:7 x 5
Inner rectangle: 22, 27

Standard input

Input begins with a single number t $(1 \leq t \leq 100)$, which denotes the number of test cases.

Each test case begins with a line, which contains 5 space-separated integers l, h, n, d_1 and d_2.

Standard output

For each test case output a single number which is the `xor` of all the number that are not common to the rectangles.

Constraints and notes

- $1 \leq t \leq 100$
- $1 \leq l, h \leq 10^6$
- $1 \leq n \leq 10^9$
- $d_1 \leq d_2$
- The sum of l for all the tests in a file is $\leq 10^6$
- The sum of h for all the tests in a file is $\leq 10^6$

Input	Output	Explanation
2 10 7 1 23 48 7 5 4 22 27	80 42	The test cases are for the images above. The `xor` of the uncommon numbers are 80 and 42 respectively.

Aeneas' cryptographic disc (4th c. B.C.)

Time limit: *1520 ms*
Memory limit: *72 MB*

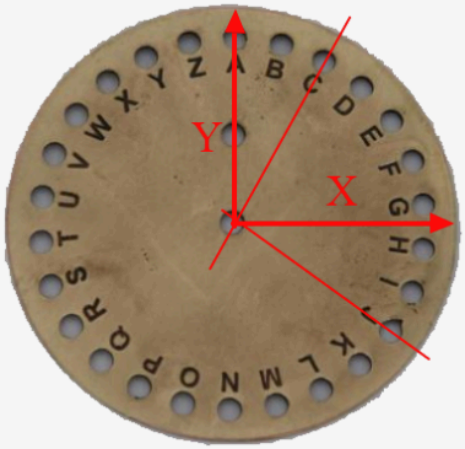
Aeneas' cryptographic disc was a bronze disc with 2424 holes (one for each letter of the Greek alphabet) on the periphery and one in the center. The sender formed the message by passing a thin thread through the holes of the relevant letters and the receiver read it by removing the thread and noting the letters from right to left.

Given the specifications of a disc created for the Latin alphabet, you are requested to calculate the length of the thread.

Assumptions

- All letters have the same and exact distance from the center of the disc, as given in the input
- The thread is so small, that its width doesn't change the result
- Each letter hole is ideal and so small that while the thread can pass successfully only at the center of the hole, it doesn't change the result, even if it passes numerous times.
- The disc is so thin, that every time the thread turns in a letter hole from front to the back of the disc and vice versa, it doesn't affect the overall result
- The thread always starts from the center of the disc and goes towards the first letter, then the second one, etc
- In case, we have multiple consecutive equal letters then only the first one is used

Standard input



The first line contains a single integer, the distance between the center of the disc and each letter (all letters have the same distance from the center), within the range $[1, 100]$ cm.

Each of the next 26 lines has one capital English letter (not necessarily in alphabetical order) followed by a decimal number, the angle in degrees of the letter from the x-axis, counterclockwise within the range $[0.0, 360.0)$. For example, in the image above, you can see letter A is on 90 degrees and C is on about 60 degrees.

Finally, the last line contains a phrase or paragraph in English, with maximum length of 10^6 characters. This phrase/paragraph may contain spaces, punctuation marks or other non English letters which you must ignore. You should use only English letters, small or capital, in order to calculate the final length of the thread, where small letters are auto transformed into capital ones.

Standard output

You should print a single integer, which is the length in cm of the thread needed to express the message given in the input as stated above as a decimal, **rounded up** to the next integer number. Thread starts from point $(0, 0)$ and ends at the final given letter.

Constraints and notes

- High precision variables might be needed for you floating point operations (e.g. in C/C++ using `double` is preferred)

-)

Input	Output
52 A 168.05 B 41.27 C 119.19 D 312.43 E 236.94 F 269.85	763

Input	Output
G 318.46 H 206.02 I 140.19 J 162.81 K 199.80 L 207.06 M 217.69 N 220.22 O 282.10 P 80.42 Q 312.29 R 324.76 S 348.38 T 311.84 U 289.66 V 137.41 W 175.23 X 0.47 Y 198.07 Z 251.39 IEEEXtreme rocks!	

Collecting Gold

Time limit: *2480 ms*
Memory limit: *264 MB*

Ariadne is the queen of Knossos in Crete and she is the ruler of all the cities and villages in the Mediterranean basin. Ariadne is in great need of gold. Her geographers put a **unique** *id* on every city/village that represents the kilos of gold that can be extracted from that city/village. The *id* is an integer k ($1 \leq k \leq 10^{18}$). The kilos of gold for every city/village can be calculated from the *id* as follows:

- Kilos = the number of unique consecutive prime numbers, starting from number 2, such that their product is less or equal than the *id* of the city/village

For example if the city *id* is 40 the kilos of gold that can be extracted is 3 $(2 * 3 * 5 = 30 \leq 40)$.

Ariadne is sending a mission to collect the gold. The mission will start from the city/village with the smallest *id* and will finish at the city/village with the biggest *id*, following one of the shortest routes between the two cities. The distance between two cities is given as it is described below in input.

There is always a route that connects the starting and finishing cities.

Standard input

The first line contains two integers n and m, where n represents the number of cities/villages and m the number of the connections between them.

Each of the next n lines contains the *id* of a city/village.

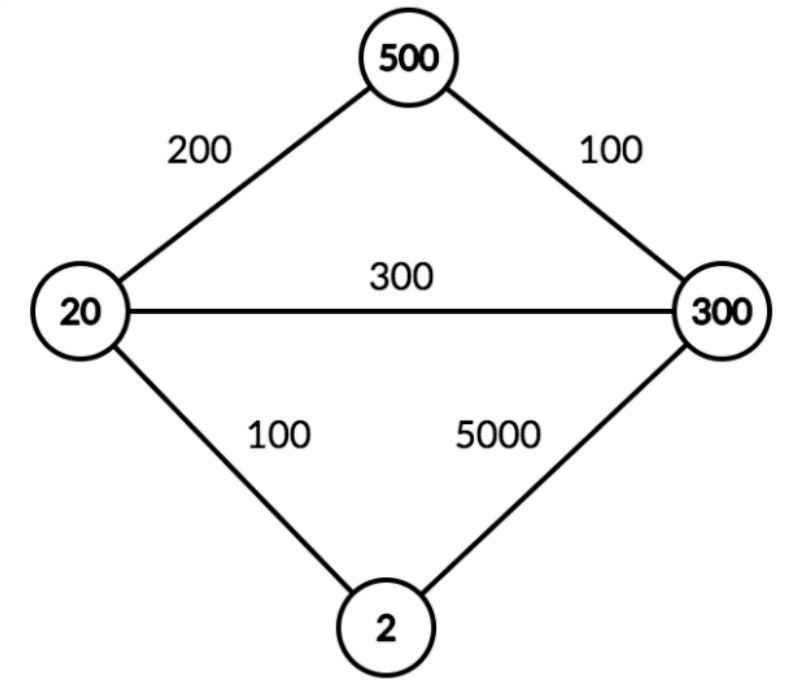
Each of the next m lines contains three integers x, y, d and represents the connection from x to y with distance d . The connections are not directed.

Standard output

One integer, the maximum kilos of gold that the mission can collect.

Constraints

- $1 \leq n \leq 2 * 10^4$
- $0 \leq m \leq 2 * 10^4$
- $1 \leq k \leq 10^{18}$, all k are unique
- $1 \leq x, y \leq 10^{18}$
- $0 < d \leq 10^5$

Input	Output	Explanation
4 5 500 2 300 20 2 300 5000 2 20 100 300 20 300 300 500 100 20 500 200	7	The cities are connected as follows.  The mission will follow the route 2 -> 20 -> 500 and will collect, 1 kilo from city 2, 2 kilos from city 20 and 4 kilos from city 500. In total 7 kilos.

Knin

Time limit: *2480 ms*
Memory limit: *14 MB*

Ariadne likes strategy board games like Chess and Go. One day she thought of her own two player strategy board game which goes as follows.

The game is played on an $w \times h$ (w for width, h for height) board which is cut into 1×1 squares. Players, starting from player 1, place in turn a piece in one square. Player pieces are **distinguishable**. More specifically player 1 pieces are labelled with a 1 and player 2 pieces are labelled with a 2.

In every $n \times n$ square (for $n > 2$) there can be at most $n - 1$ pieces (counting both players' pieces). Note that when applying this rule, the $n \times n$ square must be contained within the board and be aligned to the board grid.

The player that can't place a piece on the board on his turn loses.

Given the width w and height h of a board and the state of the board can you tell if the position is valid or not? A valid position is a position that can be reached from an empty board following the rules of the game.

Standard input

The first line of the input will contain an integer t ($1 \leq t \leq 100$), , which is the number of test cases to follow.

Each case begins with a line containing two numbers h and w ($1 \leq h, w \leq 2\,500$), the size of the board.

Each of the following h lines contains w characters each denoting the piece at each square.

- `.` symbolises an empty square
- `1` symbolises a player 1 piece
- `2` symbolises a player 2 piece

Standard output

For each test case print `YES` or `NO` depending if the input state if the board is valid or not.

Constraints and notes

- $1 \leq t \leq 100$
- $1 \leq h, w \leq 2\,500$
- The sum of $h * w$ for all the tests in a file is $\leq 2\,500^2$

Input	Output	Explanation
3 3 3 1.1 .2. 212 4 4 ...1 1... .2.. .... 1 2 ..	NO YES YES	• In the first example the 3×3 the square starting at $(0,0)$ and finishir at $(2,2)$ has 6 pieces which are more than the 2 pieces which are allowed for 3×3 square. • In the second example the state is valid. • In the third example the state is valid.

Mister Counter

Time limit: *1280 ms*
Memory limit: *136 MB*

Do you know Mr. Counter? Well, he is obsessed with counting. For instance, he counts the number of distinct configurations that a phone number might have with k digits, also the number of ways he can dress up with his current clean clothes. This time he has been thinking of a more complicated task.

For any array a of **distinct** integers, Mr. Counter denotes by F_a an array that has the following properties:

1. It contains the same elements as a
2. F_a is lexicographically greater than a
3. From all the arrays that respect property 2, F_a is the lexicographically smallest one.
4. If there no array lexicographically greater than a, F_a is equal to a.

An array a is considered smaller than another array b if and only if there is a position i that satisfies $a_1 = b_1, a_2 = b_2, ..., a_{i-1} = b_{i-1}$, $a_i < b_i$.

Mr. Counter also defines the *cost* of an array a as the number of indices where a differs from F_a.

Now it's your turn to help with a special task. Mr. Counter has an array a consisting of n **distinct** integers and he asks you q queries. Each query consists of two indices l and r $(1 \le l \le r \le n)$. Find the cost of the subarray $a_{l...r}$.

Standard input

The first line contains an integer n representing the number of elements a.

The second line contains the n elements of a.

The third line contains the number of queries q.

Each of the next q lines contains two numbers l and r $(1 \le l \le r \le n)$.

Standard output

Print the answer for each query on a different line.

Constraints and notes

- $1 \le n \le 250\,000$
- $0 \le a_i \le 10^9$
- $1 \le q \le 250\,000$

Input	Output
5 5 4 3 2 1 2 1 5 2 4	0 0

Input	Output
6 0 2 7 5 4 3 2 1 2 1 6	2 5

Fill The Pixels

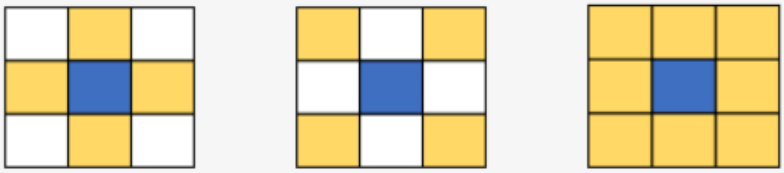
Time limit: *1280 ms*
Memory limit: *264 MB*

You were going through a bunch of black and white pictures and all of a sudden, a quirky idea struck to your mind – how many clicks does it take to convert all the white pixels into black and clear out a picture using brushes? So, you scanned and converted several pictures into binary data. Each pixel of the binary pictures are now represented by an integer, 0 for a black pixel and 1 for a white pixel.

To modify a picture, you can only use one of the three brushes each time:

- `+` (plus) - When used, the pixels at the left, top, right and bottom of the current pixel are also colored in the same color.
- `x` (cross) - When used, the pixels at the top-left, top-right, bottom-right and bottom-left of the current pixel are also colored in the same color.
- `*` (star) - When used, all 9 pixels around the current pixel are also colored in the same color.

Each of these brushes, when used, works **recursively**. So if you color a pixel, its neighbours will get colored, then the neighbours of the neighbours, and so on.



Your task is to compute the minimum clicks you will have to perform with each brush in order to clear out a picture.

So you will have to compute the minimum clicks that are needed to clear the picture using only the `+` brush, then compute the minimum clicks that are needed to clear the picture using only the `x` brush and finally compute the minimum clicks that are needed to clear the picture using only the `*` brush.

Standard input

Input begins with a single number t ($1 \leq t \leq 100$, which denotes the number of pictures to process.

Each test case begins with a line, which contains 2 space-separated integers w and h denoting the width and the height of the picture in pixels.

Following there will be h lines denoting the w pixel values (either 0 or 1) of each row of the picture.

Standard output

For each test case output, 3 space separated integers which denote the minimum number of clicks needed to clear out the picture for each of the three brushes: `+`, `x` and `*`.

Constraints and notes

- $1 \leq t \leq 100$
- $1 \leq w, h \leq 1\,000$

Input

```
2
7 4
1011001
0010001
0001000
0000001
26 13
11111111111111111111111111
11111100111111111100111111
11110001111100111110001111
11000001111000011110000011
10000000111000011100000001
10000000000000000000000001
00000000000000000000000000
00000000000000000000000000
10000000000000000000000001
10000110001000010001100001
11001111111100111111110011
11100111111100111111100111
11111111111111111111111111
```

Output

```
5 6 4
2 18 2
```

Running Up Stairs

Time limit: *1280 ms*
Memory limit: *136 MB*

Every 12-year-old boy (at least in America) feels the need to run up the stairs skipping some of them. For this problem, we are going to assume that he randomly skips single steps. (with each step of his foot, he may land on the next step or he may skip one step and land on the following step). For example, if there are three steps, he could traverse them three ways:

The challenge in this problem is: given a number of stairs, calculate the number of ways he could traverse them.

Standard input

The input will start with a single line containing one integer t $(1 \leq t \leq 5)$ specifying the number of instances of the problem. Each subsequent line will contain one instance of the problem – a single integer n $(1 \leq n \leq 22\,000)$ specifying the number of steps in the stair case.

Standard output

For each instance of the problem, your program must output one line containing a single integer – the number of ways the steps could be traversed.

Constraints and notes

- $1 \leq t \leq 5$
- $1 \leq n \leq 22\,000$

Input	Output
1 3	3
1 5	8
2 1 2	1 2

Pyramid

Time limit: *1280 ms*

Memory limit: *136 MB*

A pyramid of numbers can be represented as a matrix a of n rows, where the i^{th} row has elements only on the first i columns. In addition, $a_{i,j} = a_{i+1,j} + a_{i+1,j+1}$, for the first $n-1$ rows. For the bottom row this doesn't apply, those numbers can be any **strictly positive** integers.

Given an integer s, count the number of pyramids where $a_{1,1} = s$, for all possible values of n.

Standard input

The first line contains integer s.

Standard output

Print the number of pyramids with the given property. As this number can be very large, output its value modulo $10^9 + 7$.

Constraints and notes

- $1 \leq s \leq 10^5$

Input	Output	Explanation
4	5	There are 5 distinct pyramids with $A_{1,1} = 4$

Fibonacci

Time limit: *320 ms*

Memory limit: *264 MB*

Dr. Fibonacci is modeling the bacteria population growth using a famous sequence. Initially, a bacterium is placed inside a test tube. There will be 1 bacterium in the first minute, 2 bacteria in the second minute, 3 in the third minute, 5 in the fourth minute and so on. At the end of the minute n, the number of bacteria will be equal to the number of bacteria in minute $n-2$ plus the number of bacteria in minute $n-1$.

Dr. Xtreme wants Dr. Fibonacci to use the same model with an enhancement to model the human population growth. Dr. Xtreme wants to enhance the model by including a disaster scenario where most humans are destroyed. If there is a disaster in generation m, then only the number of humans in that generation modulo 10 will survive. You goal is develop the model with the disaster scenario, so that we can compute how many humans will survive after the disaster occurs.

Standard input

Input begins with a single number t ($1 \leq t \leq 100$), which denotes the number of test cases.

On the following t lines there will be a single integer m ($1 \leq m \leq 10^9$) which indicates the generation where the disaster occurs for the specific test case.

Please note that there will be only one disaster per test case.

Standard output

For each test case, output a single integer per line indicating the number of humans who survived the disaster.

Constraints and notes

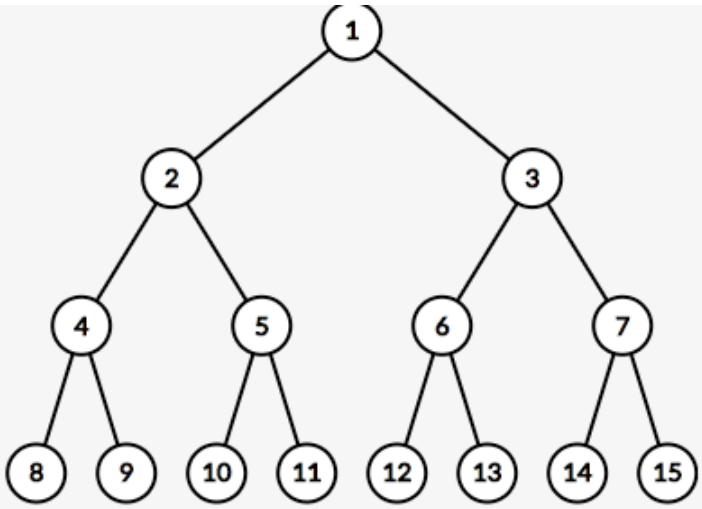
- $1 \leq t \leq 100$
- $1 \leq m \leq 10^9$

Input	Output	Explanation
3 2 4 10	2 5 9	There are three test cases. At the end of 22nd, 44th and 1010th generation, there will be 22, 55 and 8989 humans respectively. When the disaster occurs, there will be 22, 55 an

Rumour

Time limit: *1280 ms*
Memory limit: *264 MB*

Consider a full binary tree with an infinite number of levels. The vertices are labeled with positive integers, starting from the root and moving down the tree, from left to right on each level. In the example below you can see the first 4 levels of the tree:



Answer q $(1 \leq q \leq 10^6)$ queries of the form: find the minimum distance between vertices a and b $(1 \leq a, b \leq 10^{18})$.

Standard input

The first line contains integer q.

The next q lines contain two integers, a and b.

Standard output

Print q lines, on line i the answer for the i^{th} query.

Constraints and notes

- $1 \leq q \leq 10^6$
- $1 \leq a, b \leq 10^{18}$

Input	Output
4 1 1 1 2 2 3 10 6	0 1 2 5

Odd Cycle Check

Time limit: *1280 ms*

Memory limit: *136 MB*

In a distant country there are n ($1 \leq n \leq 10^4$) cities, but there aren't any roads connecting them. To make things better, the king of the country has decided to build exactly one road every day, the i^{th} road connecting two distinct cities a_i and b_i. Because the country has the technology to build bridges and tunnels, you can assume two roads never intersect.

The system of roads is *special* if it's possible to start in some city a, move along an **odd** number of roads and end up in the starting city a.

Your goal is to find the first day when the system of roads becomes special. This problem is interactive, meaning you'll have to communicate with a special program called the *interactor*. More details follow below.

Interaction

First you should read a single integer n.

Each of the following lines contains two integers a and b, representing two cities connected by a new road.

After reading every pair of integers a and b, you should output 1 if the graph is not special, or 0 if it just became special. In case you print 0, your program should stop.

Warning: You should print your answer (0 or 1) **before** reading the next line of input.

Warning 2: Don't forget to flush after every output operation!

Constraints and notes

- $3 \leq n \leq 10^4$
- It is guaranteed the system of roads will become special in the first $2 * 10^4$ days.

Interaction

```
4
1 4
```

```
1
```

```
2 1
```

```
1
```

```
3 2
```

```
1
```

```
4 2
```

```
0
```

```
4
1 2
```

```
1
```

```
1 3
```

```
1
```

```
4 2
```

```
1
```

```
3 4
```

```
1
```

```
4 1
```

```
0
```

```
5
1 2
```

```
1
```

```
2 3
```

```
1
```

```
3 4
```

```
1
```

```
4 5
```

```
1
```

```
5 1
```

```
0
```

Interaction

```
6
1 3
```

```
1
```

```
5 6
```

```
1
```

```
4 3
```

```
1
```

```
4 6
```

```
1
```

```
3 5
```

```
1
```

```
6 1
```

```
1
```

```
2 1
```

```
1
```

```
2 4
```

```
1
```

```
3 6
```

```
0
```

Lion Territory

Time limit: *1280 ms*
Memory limit: *136 MB*

In your last trip to Africa, you had the luck to encounter k lions. The savanna can be encoded as an $n \times m$ ($1 \leq n, m \leq 10^3$) matrix, and the i^{th} lion was located at cell (r_i, c_i).

The (claimed) territory of the i^{th} lion spans over all the cells at Manhattan distance at most d_i from its location. It is possible that territories overlap.

Your task it to identify the lion that is located on most of other lions' territories.

Standard input

The first line contains three integers n, m and k.

The next k lines contain r_i, c_i and d_i, representing the row, column and distance for the i^{th} lion.

Standard output

Print two integers representing the smallest index of the lion that is situated on most of other lion' territories and the number of territories.

Constraints and notes

- $1 \leq n, m \leq 10^3$
- $1 \leq k \leq n * m$
- $1 \leq r_i \leq n$
- $1 \leq c_i \leq m$
- $0 \leq d_i \leq 10^3$
- It is possible that 2 or more lions are located in the same cell

Input	Output	Explanation
5 4 4 2 1 3 3 1 2 3 4 2 2 4 2	4 2	Lion 4 is situated on the territory of lions 1 and 3.

Jarawi and The Interview

Time limit: *2480 ms*

Memory limit: *264 MB*

Today Jarawi had a job interview for a very important tech company, the interviewer asked him to solve the next problem:

Given a string s $(1 \leq |s| \leq 10^6)$, answer q $(1 \leq q \leq 5 * 10^4)$ queries. For each query he received a string p_i $(1 \leq |p_i| \leq 100)$ and he had to determine the size of the longest suffix of p_i which is a **subsequence** of s (a suffix of a string is a **substring** that occurs at the end).

Unfortunately Jarawi could not solve the task, so he wants to know if you can solve it.

Standard input

The first line contains a string s.

The second line contains an integer q.

Each of the next q lines contains a string p_i.

Standard output

For each query you need to print the size of the longest suffix of p_i which is a subsequence of s.

Constraints and notes

- $1 \leq |s| \leq 10^6$
- $1 \leq q \leq 5 * 10^4$
- $1 \leq |p_i| \leq 100$
- All strings contain only lower case letters of the English alphabet.

Input	Output	Explanation
xaybaba 2 aaba yx	4 1	The first query is "aaba" which has the next suffixes {"a","ba","aba","aaba"}, all of them are subsequence of "xaybaba", so the longest has size 4. The second query is "yx" which has the next suffixes {"x", "yx"}, only "x" is subsequence of "xaybaba" and it has size 1.

Maximum Sum

Time limit: *1280 ms*

Memory limit: *264 MB*

Given a sequence that contains n ($1 \leq n \leq 10^5$) integers, represented by $a_1, a_2, ..., a_n$ ($0 \leq a_i \leq 10^5$), you have the amazingly easy task to reorder the sequence and maximize the formula $\sum_{i=1}^{n-1} a_i * a_{i+1}$.

Standard input

The first line of the input will contain an integer t ($1 \leq t \leq 25$), which is the number of test cases to follow.

Each test case is composed of two lines:

- The first line of each test case contains an integer n, which is the length of the sequence.
- The second line of the test case contains n integers indicating the $a_1, a_2, ..., a_n$ sequence, each integer is separated by one space.

Standard output

The output should contain $2 * t$ lines, two lines for each test case:

- The first line of each test case should contain the maximum sum.
- The second line of the test case should contain n integers indicating the reordered sequence. If there is more than one solution, print the one with the smallest alphabetical order.

Contraints and notes

- $1 \leq t \leq 25$
- $1 \leq n \leq 10^5$
- $0 \leq a_i \leq 10^5$
- Sequence a is considered smaller than sequence b if and only if there is a position i that satisfies $a_1 = b_1, a_2 = b_2, ..., a_{i-1} = b_{i-1}, a_i < b_i$.

Input	Output
4 10 1 2 1 5 4 2 1 1 8 9 3 5 2 0 9 5 9 0 0 4 6 2 3 2 4 2 1 7 1	173 1 1 1 2 4 8 9 5 2 1 10 0 2 5 152 0 0 2 3 5 9 6 4 2 23 1 2 7 1

BeetleBag

Time limit: *1280 ms*
Memory limit: *38 MB*

Beetleman joined the Strangers, a team of super heroes who protect cyber world.

In order to increase Beetleman's fighting power, Copperman allowed Beetleman to choose gadgets from his labs freely.

However, Beetleman has limited space in his hero bag.

Your task is to help Beetleman choose gadgets to increase his fighting power as much as possible.

Standard input

The first line of input has one integer t ($1 \leq t \leq 25$), the number of test cases that will follow.

For each t there will be a line that contains two integers, number c ($1 \leq c \leq 500$), the capacity of Beetleman's bag, and number n ($1 \leq n \leq 200$), the number of gadgets in Copperman labs.

Then for each above line, there will be n lines that will contain two integers, the number w ($1 \leq w \leq 100$), the gadget's weight and the number f ($1 \leq f \leq 1\,000$), the fighting power of the gadget.

Standard output

Output will have t lines containing the maximum fighting power from Copperman's gadgets that can fit into Beetleman's bag.

Constraints and notes

- $1 \leq t \leq 25$
- $1 \leq c \leq 500$
- $1 \leq n \leq 200$
- $1 \leq w_i \leq 100$ for $1 \leq i \leq n$
- $1 \leq f_i \leq 1\,000$ for $1 \leq i \leq n$

Input	Output	Explanation

```
2
6 2
1 17
6 15
5 5
1 1
2 2
3 3
4 4
5 5
```

```
17
5
```

Input explanation

```
1  2    <<<< two test
       cases
2  6 2  <<<< the first
       test case has 6
       capacity and 2
       gadgets to choose
       from.
3  1 17 <<<< weight 1,
       fighting power 17
4  6 15 <<<< weight 6,
       fighting power 15
5  5 5  <<<< the second
       test case has 5
       capacity and 5
       gadgets to choose
       from.
6  1 1  <<<< weight 1,
       fighting power 1
7  2 2  <<<< weight 2,
       fighting power 2
8  3 3  <<<< weight 3,
       fighting power 3
9  4 4  <<<< weight 4,
       fighting power 4
10 5 5  <<<< weight 5,
       fighting power 5
11
```

Output explanation

```
1  17 <<<< maximum
       fighting power from
       first test case
2  5  <<<< maximum
       fighting power from
       second test case
3
```