ACM-ICPC VIETNAM NATIONAL ONLINE TESTING SESSION

Hanoi, 20th October 2013

Problem Set Overviews

ID	Name	Time Limit
A	Roll Dice	10s
B	Scale Icon	10s
C	Do Chores	10s
D	Firehose	10s
E	Primed Sequence	10s
F	Number Matrix	10s
G	Segments	10s
H	Ball Machine	10s
I	Brunhilda's Birthday	10s

A. ROLL DICE

There are 2 dice on the table. One die has m sides labelled 1, 2, 3, …, m. The other has n sides labelled 1, 2, 3, …, n.

Write a program to determine how many ways to roll the dice to get the sum 10.

For example, when m = 6 and n = 8, we have 5 ways :
2 + 8 = 10
3 + 7 = 10
4 + 6 = 10
5 + 5 = 10
6 + 4 = 10

Input

The first line contains an integer T denotes the number of test cases.
Following T lines, each contains 2 integers : m, n respectively.

Output

T lines, on each line print out the number of ways 10 may be rolled on these two dice (K), with following format :

```
There are K ways to get the sum 10.
```

If K = 1:

```
There is 1 way to get the sum 10.
```

Note that in the output, the word "way" should be used if there is only one way to achieve the sum of 10; otherwise, the word "ways" should be used in the output.

Constraints

T ≤ 50
1 ≤ m, n ≤ 1000

Example

Input	Output
2 6 8 12 4	There are 5 ways to get the sum 10. There are 4 ways to get the sum 10.

B. SCALE ICON

You have been asked to take a small icon that appears on the screen of a smart telephone and scale it up so it looks bigger on a regular computer screen.

The icon will be encoded as characters (x and *) in a 3 × 3 grid as follows:

```
*x*
 xx
*  *
```

Write a program that accepts a positive integer scaling factor and outputs the scaled icon. A scaling factor of k means that each character is replaced by a k × k grid consisting only of that character.

Input

The first line contains an integer T denotes the number of test cases. Each of the next T lines contains a positive integer k such that k < 25.

Output

The output of each test case will be 3k lines, which represent each individual line scaled by a factor of k and repeated k times. A line is scaled by a factor of k by replacing each character in the line with k copies of the character.

Example

Input	Output

1
3
	xxx
	xxx
	xxx
	  xxxxxx
	  xxxxxx
	  xxxxxx
	*** ***
	*** ***
	*** ***

C. DO CHORES

You have been asked by a parental unit to do your chores.

Each chore takes a certain amount of time, but you may not have enough time to do all of your chores, since you can only complete one chore at a time. You can do the chores in any order that you wish.

What is the largest amount of chores you can complete in the given amount of time?

Input

The first line contains the number of test cases.

Each test case have following format:

The first line consists of an integer T ($0 \leq T \leq 100000$), which is the total number of minutes you have available to complete your chores.

The second line consists of an integer C ($0 \leq C \leq 100$), which is the total number of chores that you may choose from.

The next C lines contain the (positive integer) number of minutes required to do each of these chores. You can assume that each chore will take at most 100000 minutes.

Output

The output for each test case will be the maximum number of chores that can be completed in time T .

Example

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

D. FIREHOSE

There is a very unusual street in your neighbourhood. This street forms a perfect circle, and the circumference of the circle is 1,000,000. There are H ($1 \le H \le 1000$) houses on the street. The address of each house is the clockwise arc-length from the northern-most point of the circle. The address of the house at the northern-most point of the circle is 0.

You also have special firehoses which follow the curve of the street. However, you wish to keep the length of the longest hose you require to a minimum.

Your task is to place K ($1 \le K \le 1000$) fire hydrants on this street so that the maximum length of hose required to connect a house to a fire hydrant is as small as possible.

Input

The first line contains the number of test cases.

Each test case has following format:

The first line will be an integer H, the number of houses. The next H lines each contains one integer, which is the address of that particular house, and each house address is at least 0 and less than 1,000,000. On the H + 2nd line is the number K, which is the number of fire hydrants that can be placed around the circle. Note that a fire hydrant can be placed at the same position as a house. You may assume that no two houses are at the same address.

Output

Each test case, output one line, contains the length of hose required so that every house can connect to its nearest fire hydrant with that length of hose.

Example

Input	Output
1 4 0 67000 68000 77000 2	5000

E. PRIMED SEQUENCE

Given a sequence of positive integers of length n, we define a primed subsequence as a consecutive subsequence of length at least two that sums to a prime number greater than or equal to two.

For example, given the sequence:

3, 5, 6, 3, 8

There are two primed subsequences of length 2 (5 + 6 = 11 and 3 + 8 = 11), one primed subsequence of length 3 (6 + 3 + 8 = 17), and one primed subsequence of length 4 (3 + 5 + 6 + 3 = 17).

Find the shortest primed subsequence or announce that there's no such sequence. If there are several solutions, print the one that occurs first (i.e index of the start element is smallest).

Input

Input consists of a series of test cases. The first line consists of an integer t ($1 \leq t \leq 20$), the number of test cases.

Each test case consists of one line. The line begins with the integer n, $0 < n \leq 10000$, followed by n non-negative numbers less than 10000 comprising the sequence.

Output

For each sequence, print the "Shortest primed subsequence is length x:", where x is the length of the shortest primed subsequence, followed by the shortest primed subsequence, separated by spaces (exact one space between 2 consecutive element). Note that there is also one space between the ":" and the first element of the sequence. There must not be any spaces at the end of the line.

If there are multiple such sequences, print the one that occurs first.

If there are no such sequences, print "This sequence is anti-primed.".

Example

Input
3
5 3 5 6 3 8
5 6 4 5 4 12
21 15 17 16 32 28 22 26 30 34 29 31 20 24 18 33 35 25 27 23 19 21

Output
Shortest primed subsequence is length 2: 5 6
Shortest primed subsequence is length 3: 4 5 4
This sequence is anti-primed.

F. NUMBER MATRIX

Johnny Cannook has been trapped in the matrix: no, not that matrix. This matrix is a grid of width N ($1 \le N \le 100$) numbers (in the range 0 to 9) in each row, and M rows ($1 \le M \le 100$).

Johnny can pick any position on the first row to begin at. He must make it to row M of the matrix in order to escape.

However, there is a restriction. Johnny can only choose a trinity of numbers (from the range 0 to 9), and he can only step on those positions which are one of these three chosen numbers. That is just the way it is.

The path may begin at any position in row 1, and can move left, right, up, or down (no diagonal movement allowed) to any number, so long as that number is in the set of the chosen three.

Input

The first line contains the number of test cases.

Each test case has following format:

The first line contains the two integers N and M. On the next M lines, there are N numbers (each separated by a space).

Output

The output for each test case is one line long, containing three integers: the trinity of numbers that Johnny should chose in order to escape the matrix.

If there is no path from row 1 to row M (that is, Johnny is stuck in the matrix FOREVER), the output should be three −1.

If there is a path, then the lexicographically first three numbers should be outputted. (Notice that 0, 0, 0 comes before 0,0,1, which comes before 0,0,2, ..., which comes before 9,9,8 which comes before 9, 9, 9). You should notice that the three chosen numbers need not be distinct.

Example

Input	Output
1 6 5 0 1 2 3 4 5 0 0 0 1 1 1 0 1 2 3 3 4 5 1 4 1 9 4 9 5 6 2 4 6	0 1 5

G. SEGMENTS

You are to find the length of the shortest path from the top to the bottom of a grid covering specified points along the way.

More precisely, you are given an n by n grid, rows 1..n and columns 1..n ($1 \le n \le 20000$). On each row i, two points L(i) and R(i) are given where $1 \le L(i) \le R(i) \le n$. You are to find the shortest distance from position (1, 1), to (n, n) that visits all of the given segments in order. In particular, for each row i, all the points

(i, L(i)), (i, L(i) + 1), (i, L(i) + 2), ..., (i, R(i)),

must be visited. Notice that one step is taken when dropping down between consecutive rows. Note that you can only move left, right and down (you cannot move up a level). On finishing the segment on row n, you are to go to position (n,n), if not already there. The total distance covered is then reported.

Input

The first line contains the number of test cases. Each test case has following format:

The first line consists of an integer n, the number of rows/columns on the grid. On each of the next n lines, there are two integers L(i) followed by R(i) (where $1 \le L(i) \le R(i) \le n$).

Output

The output for each test case is one integer, which is the length of the (shortest) path from (1, 1) to (n, n) which covers all intervals L(i), R(i).

Example

Input	Output
1 6 2 6 3 4 1 3 1 2 3 6 4 5	24

Explanation :

On the first row, we must traverse 5 units to the right and then drop down one level.On the second row, we must traverse 3 units to the left and drop down one level. On the third row, we must traverse 2 units to the left and drop down one level. On the fourth row, we move 1 unit to the right and then drop down one level. On the fifth row, we move 4 units to the right and drop down one level.

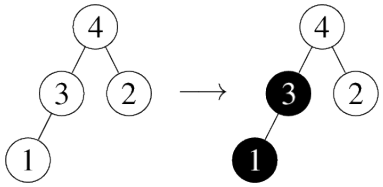
On the sixth (and final) row, we move 2 units left, then 2 units right. In total, we have moved 6 + 4 + 3 + 2 + 5 + 4 = 24 units.

H. BALL MACHINE

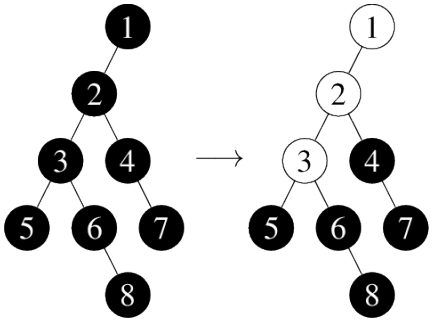
We have a "ball machine" that can be visualized as a rooted tree. The nodes of the tree are numbered from 1 to N. Each node is either empty or contains one ball. Initially, all nodes are empty. While running, the machine can perform operations of two different types:

1. Add k balls to the ball machine: Balls are put one by one into the root node. As long as a ball has an empty node directly beneath it, it will roll down. If there are multiple empty child nodes, the ball will choose the one which has the node with the smallest number in its subtree. So if the ball rolls down multiple levels, it makes a choice at each level.

For example: If we add two balls to the machine in the picture below, they will go to nodes 1 and 3: The first ball rolls from node 4 to node 3 because node 3 is empty and it contains node 1 in its subtree (which consists of nodes 3 and 1); it further rolls from node 3 to node 1. The second ball rolls from node 4 to node 3 as well and stops there.



2. Remove a ball from a specified node: This node becomes empty and balls from above (if there are any) roll down. Whenever a parent of an empty node contains a ball, this ball will roll down. If we remove balls from nodes 5, 7 and 8 (in this order) from the machine in the picture below, nodes 1, 2 and 3 will become empty.



Input

The first line contains the number of test cases. Each test case has following format:

The first line contains two integers N and Q – the number of tree nodes and the number of operations.

The next N lines describe the ball machine. Each of these lines contains one integer, the number of a node: the i-th of these lines contains the number of node i's parent node, or 0 if node i is the tree root.

Each of the next Q lines contains two integers and describes an operation to be performed. An operation of type 1 is denoted by 1 k where k is the number of balls to be added to the machine. An operation of type 2 is denoted by 2 x where x is the number of

the node from which a ball is to be removed. It is guaranteed that all performed operations are correct: Operations will not require to add more balls than there are empty nodes in the ball machine or to remove a ball from an empty node.

Output

Output for each test case has following format:

For each operation of type 1, output the number of the node where the last inserted ball ended up. For each operation of type 2 output the number of balls that rolled down after removing the ball from the specified node.

Constraints

N, Q ≤ 100 000.

Example

Input	Output
1 8 4 0 1 2 2 3 3 4 6 1 8 2 5 2 7 2 8	1 3 2 2

I. BRUNHILDA'S BIRTHDAY

Except for her affinity towards old armours, Brunhilda is a normal seven year old girl. Thus, she is planning the perfect birthday party, for which she has invented the following game:

All children run around until some number k is announced. Then all children try to form groups of exactly k people. As long as at least k children are left over, further groups of k children are formed. In the end, less than k children are left over and will be eliminated (not literally of course) from the game. The game continues with further numbers announced, and ends if all children are out.

Brunhilda asked her father Wotan to announce the numbers in the game. Wotan does not like this game and announced ∞ when they first tried it. Brunhilda thinks this would be quite embarrassing at the party, and so she gave him a list of m prime numbers from which he can choose for each call; he may use the same number more than once.

Wotan would like to end the game as soon as possible since he has tickets for a match of his favourite football club FC Asgard. Unfortunately, Brunhilda does not know the number of children at her party yet. Now, for Q different numbers $n_1, \ldots, n_Q$ of children, Wotan wants to know in advance the least number of calls he will need to end the game.

Input

The first line of the input contains the number of test case. Each test case has the following format:

The first line contains the integers m and Q described above. The second line contains m different prime numbers p_i ($1 \le i \le m$) in ascending order: the list of prime numbers Wotan can use. The following Q lines contain one integer n_j ($1 \le j \le Q$) each: the number of children who might take part in the game.

Output

The output should consist of Q lines. The j-th line should contain the answer for n_j: if Wotan can end the game it should contain the least number of calls he needs (an integer), otherwise the line should contain the string oo (two lower case letters o, meaning ∞).

Constraints

$1 \le m \le 100000$, $1 \le Q \le 100000$, $2 \le p_i \le 10000000$, $1 \le n_j \le 10000000$

Example

Input	Output
1 2 2 2 3 5 6	3 oo