ACM-ICPC Indonesia National Contest 2010

Problem A

The Best Team

Time Limit: 2s

ACM-ICPC 2010 is drawing near and your university want to select three out of N students to form the best team. The university however, has a limited budget, so they can only afford to send one team. The coach wants his best team to be the best in term of compatibility. The compatibility of a team which member is student A, B and C is calculated by $P_{A,B} * P_{A,C} * P_{B,C}$, where $P_{i,j}$ is the compatibility of student i and student j.

Given $P_{i,j}$ for all pair of students, calculate the highest compatibility value that can be achieved by any team of three students.

Input

The first line of input contains an integer T ($1 \le T \le 100$) the number of cases. Each case begins with an integer N ($3 \le N \le 50$) the number of students. The next N lines each contains N integers $P_{i,j}$ ($0 \le P_{i,j} \le 100$). The i^{th} line and j^{th} integer denotes the compatibility of student i with student j. You may assume $P_{i,j} = P_{j,i}$ and $P_{i,i} = 0$.

Output

For each test case, output in a line the highest compatibility value that can be achieved by any team of three students.

Sample Input

```
2
3
0 2 5
2 0 8
5 8 0
4
0 3 5 4
3 0 2 6
5 2 0 8
4 6 8 0
```

Output for Sample Input

```
80
160
```

Explanation for the 1st sample input.

There are only three students so the coach has no choice but to select them.

Explanation for the 2nd sample input.

The best combination would be choosing student 1, 3 and 4.

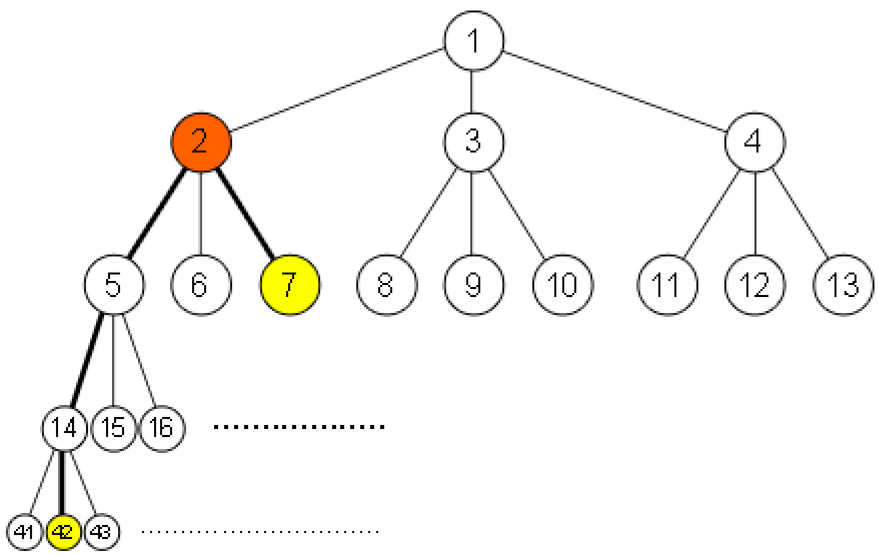
Problem B

Largest Labeled Common Ancestor

Time Limit: 2s

Given a labeled complete k-ary tree, find the largest labeled common ancestor of two given nodes. In a complete k-ary tree, the node in the tree is labeled sequentially from the left most child to the right most child, level by level. Largest labeled common ancestor of A and B is defined as the largest labeled node in the tree which has both A and B as descendants. A node is a descendant of itself.

For example, given a k-ary tree with k = 3, the largest labeled common ancestor of node 42 and 7 is node 2 (see the picture below).



Input

The first line of input contains an integer T (1 ≤ T ≤ 100) the number of cases. Each case contains three integers K, A and B (2 ≤ K ≤ 100; 1 ≤ A, B ≤ 2,000,000,000).

Output

For each test case, output in a line the largest labeled common ancestor node.

Sample Input

Output for Sample Input

```
4
3 42 7
2 2 6
2 1 105
4 10 13
```

```
2
1
1
3
```

ACM-ICPC Indonesia National Contest 2010

Problem C

Stack Machine Simulator

Time Limit: 2s

A stack machine has two type of operations:

- Push an element into stack. This operation is denoted by a command string "`+?`" where `?` is any alphabet 'a'-'z' or 'A'-'Z', it means append a character `?` into stack.
- Reverse all element in stack. This operation is denoted by a command string "`^`". It will reverse all elements in stack, do nothing if the stack is empty.

For example, command string "`+a+b+c+d^`" means: push a, push b, push c, push d and reverse. After "`+a+b+c+d^`" executed, the stack contains "`dcba`". Write a program which take the command string as the input and output the stack's content after the command is executed.

Input

The first line of input contains an integer T ($1 \le T \le 1{,}000$) the number of cases. Each case contains a string S denoting the command string as described above. S will be between 1 and 100 characters long.

Output

For each test case, output in a line the content of the stack after the command is executed.

Sample Input

```
2
+a+b
+A+B^+c^
```

Output for Sample Input

```
ab
cAB
```

ACM-ICPC Indonesia National Contest 2010

Problem D

Sum to Zero

Time Limit: 10s

Given three sets of integers of the same size: A, B and C. Your task is to calculate how many triplet {i, j, k} such that $A_i + B_j + C_k = 0$.

Input

The first line of input contains an integer T (1 ≤ T ≤ 100) the number of cases. Each case begins with an integer N (1 ≤ N ≤ 2,000) the size of each set of integer. The next three lines each contains N integers range from -1,000,000,000 to 1,000,000,000. Each line represent A, B and C respectively.

Output

For each test case, output in a line the number of different triplets.

Sample Input

```
2
2
0 3
-1 5
0 1
4
2 7 -3 9
-1 -5 4 6
-2 -7 8 -3
```

Output for Sample Input

```
1
3
```

Explanation for the 1st sample input.

There is only one triplet that sums to zero: {0, 0, 1} which corresponds to 0 + (-1) + 1 = 0.

Explanation for the 2nd sample input.

There are three triplets that sum to zero:

- {1, 1, 0} which corresponds to 7 + (-5) + (-2) = 0.
- {2, 1, 2} which corresponds to (-3) + (-5) + 8 = 0.
- {2, 3, 3} which corresponds to (-3) + 6 + (-3) = 0.

ACM-ICPC Indonesia National Contest 2010

Problem E

Playing with Boxes

Time Limit: 3s

You may be don't know about this, but Ceemot does like to play with boxes during her free time. She has N different boxes in one pile. First, she splits them into two piles (not necessary the same size), then she picks one of the piles with at least two boxes and splits it into two again. She repeats this until each pile has only one box.

As a computer scientist, she wonders the number of different ways in which she can do this.

For example, if she begins with a pile of 3 boxes (A, B and C) then there are three ways to do her weird hobby:

1. Split ABC into A and BC, split BC into B and C.
2. Split ABC into B and AC, split AC into A and C.
3. Split ABC into C and AB, split AB into A and B.

If she begins with a pile of 4 boxes (A, B, C and D) then there are eighteen ways to do this:

- Split ABCD into AB and CD, split AB into A and B, split CD into C and D.
- Split ABCD into AB and CD, split CD into C and D, split AB into A and B.
- Split ABCD into AC and BD, split AC into A and C, split BD into B and D.
- ...
- Split ABCD into A and BCD, split BCD into B and CD, split CD into C and D.
- Split ABCD into A and BCD, split BCD into C and BD, split BD into B and D.
- Split ABCD into A and BCD, split BCD into D and BC, split BC into B and C.
- ...

Help her to count the number of different ways in which she can carry out this splitting procedure. As the number may be very big, modulo the output with 1,000,000,007.

Input

The first line of input contains an integer T (1 ≤ T ≤ 1,000) the number of cases. Each case begins with an integer N (2 ≤ N ≤ 100,000) the number of different boxes Ceemot has in one pile originally.

Output

For each test case, output in a line the number of different ways in which she can carry out her splitting procedure. Modulo this number with 1,000,000,007.

Sample Input

```
3
3
4
7
```

Output for Sample Input

```
3
18
56700
```

ACM-ICPC Indonesia National Contest 2010

Problem F

Searching in Tree

Time Limit: 8s

Given a rooted tree which contains N nodes numbered from 0 to N-1 with node 0 as the root. Each node has a distinct value assigned to it. You have to write a program that can find the k^{th} smallest value in the subtree of a given node X_i.

Input

The first line of input contains an integer T ($1 \le T \le 20$) the number of cases. Each case begins with two integers N and Q ($1 \le N \le 100{,}000$; $1 \le Q \le 10{,}000$) denoting the number of nodes and queries respectively. The next line contains N integers V_i ($0 \le V_i \le 2^{31}-1$) denoting the value assigned to i^{th} node. The next line contains N integers P_i ($0 \le P_i < N$) denoting the parent of i^{th} node. Parent of node 0 will be 0. The next Q lines each contains two integers k_i and X_i ($1 \le k_i \le$ sizeof-subtree; $0 \le X_i < N$).

Output

For each test case, output "Case #X:" (without quotes) where X is the case number starting with 1. Answer each query of the cases in a separate line.

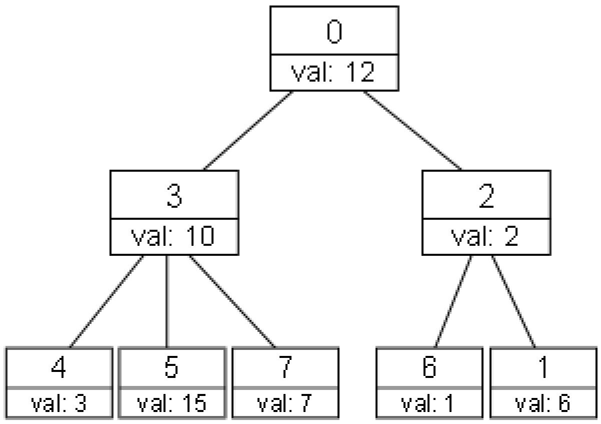
Sample Input

```
1
8 5
12 6 2 10 3 15 1 7
0 2 0 0 3 3 2 3
1 0
3 0
2 3
1 6
3 3
```

Output for Sample Input

```
Case #1:
1
3
7
1
10
```

Explanation for the 1st sample input:


0
val: 12
3
val: 10
2
val: 2
4
val: 3
5
val: 15
7
val: 7
6
val: 1
1
val: 6

ACM-ICPC Indonesia National Contest 2010

Problem G

Finding The Right Song

Time Limit: 8s

Silcat has N songs in her digital library. Each song is described as a sequence of M melodies. A melody is an integer with value between 0 and 1,000,000.

Everytime Silcat hear an interesting sequence of melodies of length 10, Silcat always asks Sucat whether she has a song in her library which contains that melodies. A sequence of melodies S exists in a song if the song has a continuous subsequence of melodies S. Silcat has ensured that any sequence of melodies of length 10 does not exist in more than one song in her library and occurs only once in a song.

Since Sucat is busy with his work today, he asks you to create a program to answer all Silcat's questions.

Input

The first line begins with two integers N and Q ($1 \le N \le 2,000$; $1 \le Q \le 20,000$) the number of songs in her digital library and the number of questions Silcat will ask consecutively. The next N lines each begins with an integer M_i ($10 \le M_i \le 1,000$) the number of melodies in i^{th} song, followed by M_i integers denoting the sequence of melodies in i^{th} song. The next Q lines each will contain 10 integers denoting the sequence of melodies S that is asked by Silcat.

Output

Answer each question in a line. If S can be found in a song, output two numbers, the song number that contains S and the starting position of the sequence S in the song (starting with 1). If S can not be found, output "not found" (without quote).

Sample Input

```
7 5
17 13 15 9 16 16 3 18 11 5 11 6 18 1 6 14 18 6
18 14 9 8 2 10 10 16 6 11 6 16 10 20 8 5 15 6 2
17 7 6 14 9 9 7 5 13 5 17 17 9 19 9 5 2 16
19 7 6 19 13 18 12 16 2 12 13 7 2 20 17 11 8 5 6 12
20 10 13 13 9 9 10 14 15 16 9 4 9 5 5 13 8 20 13 7 7
18 16 13 13 11 10 18 12 17 18 6 17 4 3 19 9 16 4 10
20 11 2 4 19 13 8 16 17 11 18 15 5 3 9 11 19 3 19 6 18
10 18 12 17 18 6 17 4 3 19
13 13 11 10 18 12 17 18 6 17
11 12 17 7 8 5 6 8 3 16
```

Output for Sample Input

```
6 5
6 2
not found
5 8
2 7
```

15 16 9 4 9 5 5 13 8 20
16 6 11 6 16 10 20 8 5 15

ACM-ICPC Indonesia National Contest 2010

Problem H

Disconnected Graph

Time Limit: 10s

Given an undirected connected simple graph of N nodes and E edges. There are up to Q nodes selected from N nodes that are special nodes. Suppose that R selected edges are destroyed, your task is to calculate how many pairs of node A-B where A and B belong to special nodes, are disconnected (there is no path from A to B or vice versa after R edges are destroyed).

Input

The first line of input contains an integer T ($1 \le T \le 50$) the number of cases. Each case begins with four integers N, E, Q and R ($1 \le Q \le N \le 50{,}000$; $0 \le R \le E \le 200{,}000$) denoting the number of nodes, edges, special nodes, and destroyed edges respectively. The nodes are numbered from 1 to N. The next E line each contains two integers A_i and B_i ($1 \le A_i, B_i \le N$) representing an edge connecting node A and node B. The next line contains Q integers P_i ($1 \le P_i \le N$) denoting the selected special nodes. The next line contains R integers T_i ($1 \le T_i \le E$) denoting the edges that are destroyed (the edges are numbered from 1 to E based on input order).

Output

For each case, output in a single line a number denoting the number of disconnected pair of special nodes.

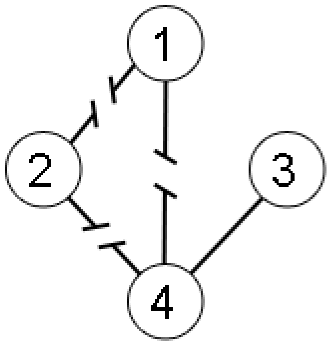
Sample Input

```
2
4 4 4 3
1 2
1 4
2 4
3 4
1 2 3 4
1 2 3
2 1 1 1
1 2
2
1
```

Output for Sample Input

```
5
0
```

Explanation for the 1st sample input



There are 5 pairs of special nodes that are disconnected: 1-2, 1-3, 1-4, 2-3 and 2-4.

Explanation for the 2nd sample input

There is only one special node, so we can't have any pair of them.

Problem I

Maximum Sum in Matrix

Time Limit: 3s

Given a matrix A of size N x M, write a program to find p_1, p_2, ..., p_n to maximize $A[1][p_1] + A[2][p_2] + ... + A[n][p_n]$. You are only allowed to change p_i no more than K times (changes occur when p_i is not equal to p_{i-1}). You may choose p_1 anywhere from 1 to M and it will not be counted as a change.

For example, given a matrix of 7 x 4 as shown below and K = 3.

	1	2	3	4			
1	3	8	5	2	P1	2	
2	12	1	4	10	P2	4	change
3	1	5	6	7	P3	4	
4	2	3	4	3	P4	4	
5	1	7	2	9	P5	4	
6	4	20	3	1	P6	2	change
7	5	3	15	8	P7	3	change

The maximum sum that you can get is 8 + 10 + 7 + 3 + 9 + 20 + 15 = 72. But if you are only allowed to make changes at most one time (K = 1), then the maximum sum that you can get is 59.

	1	2	3	4			
1	3	8	5	2	P1	2	
2	12	1	4	10	P2	2	
3	1	5	6	7	P3	2	
4	2	3	4	3	P4	2	
5	1	7	2	9	P5	2	
6	4	20	3	1	P6	2	
7	5	3	15	8	P7	3	change

Input

The first line of input contains an integer T (1 ≤ T ≤ 100) the number of cases. Each case begins with three integers N, M and K (1 ≤ N ≤ 100; 1 ≤ M ≤ 10; 0 ≤ K < N). The next N lines each contains M integers representing the given matrix. Each integer in the matrix is between 1 and 1,000 inclusive.

Output

For each case, output in a single line the maximum sum that you can get from the matrix.

Sample Input

```
2
7 4 3
3 8 5 2
12 1 4 10
1 5 6 7
2 3 4 3
1 7 2 9
4 20 3 1
5 3 15 8
7 4 1
3 8 5 2
12 1 4 10
1 5 6 7
2 3 4 3
1 7 2 9
4 20 3 1
5 3 15 8
```

Output for Sample Input

```
72
59
```