A-An Olympian Math Problem

Time Limit: 2000/1000 MS (Java/Others)   Memory Limit: 65536K

Problem Description

Alice, a student of grade 6, is thinking about an Olympian Math problem, but she feels so despair that she cries. And her classmate, Bob, has no idea about the problem. Thus he wants you to help him. The problem is:

We denote $k!$: $k! = 1 \times 2 \times \cdots \times (k-1) \times k$

We denote S : $S = 1 \times 1! + 2 \times 2! + \cdots + (n-1) \times (n-1)!$

Then S module n is ____________

You are given an integer n. You have to calculate S modulo n.

Input

The first line contains an integer T($T \leqslant 1000$), denoting the number of test cases. For each test case, there is a line which has an integer n.

It is guaranteed that $2 \leqslant n \leqslant 10^{18}$.

Output

For each test case, print an integer S modulo n.

Sample Input

```
2
2
3
```

Sample Output

```
1
2
```

Hint

The first test is: $S = 1 \times 1! = 1$, and 1 modulo 2 is 1.

The second test is: $S = 1 \times 1! + 2 \times 2! = 5$, and 5 modulo 3 is 2.

B-The writing on the wall

Time Limit: 4000/2000 MS (Java/Others) Memory Limit: 262144K

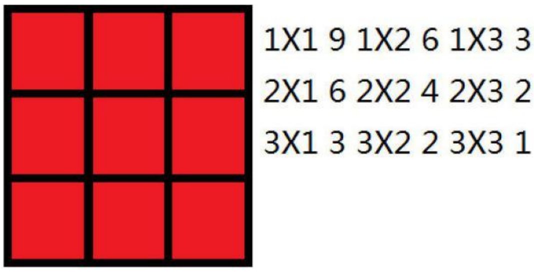
Problem Description

Feeling hungry, a cute hamster decides to order some take-away food (like fried chicken for only 30 Yuan).

However, his owner CXY thinks that take-away food is unhealthy and expensive. So she demands her hamster to fulfill a mission before ordering the take-away food. Then she brings the hamster to a wall.

The wall is covered by square ceramic tiles, which can be regarded as a n*m grid. CXY wants her hamster to calculate the number of rectangles composed of these tiles.

For example, the following 3*3 wall contains 36 rectangles:



Such problem is quite easy for little hamster to solve, and he quickly manages to get the answer.

Seeing this, the evil girl CXY picks up a brush and paint some tiles into black, claiming that only those rectangles which don't contain any black tiles are valid and the poor hamster should only calculate the number of the valid rectangles. Now the hamster feels the problem is too difficult for him to solve, so he decides to turn to your help. Please help this little hamster solve the problem so that he can enjoy his favorite fried chicken.

Input

There are multiple test cases in the input data.

The first line contains a integer T : number of test cases. $T \le 5$.

For each test case, the first line contains 3 integers n, m, k, denoting that the wall is a $n \times m$ grid, and the number of the black tiles is k.

For the next k lines, each line contains 2 integers: $x\ y$,denoting a black tile is on the x-th row and y-th column. It's guaranteed that all the positions of the black tiles are distinct.

For all the test cases, $1 \le n \le 10^5, 1 \le m \le 100, 0 \le k \le 10^5$, 1 ≤ x ≤ n, 1≤ y ≤ m.

It's guaranteed that at most 2 test cases satisfy that $n \ge 20000$.

Output

For each test case, print "Case #x: ans" (without quotes) in a single line, where x is the test case number and ans is the answer for this test case.

Sample Input

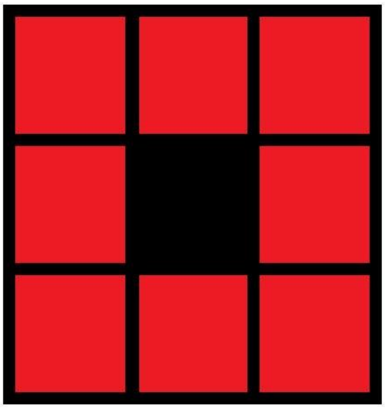
2
3 3 0
3 3 1
2 2

Sample Output

Case #1: 36
Case #2: 20

Hint

The second test case looks as follows:



C-GDY

Time Limit: 4000/2000 MS (Java/Others)   Memory Limit: 65536K

Problem Description

Feeling bored, a group of hamsters decide to play a kind of card game named “GDY”.

“GDY” is a kind of card game. To begin with, we pile up m cards with a number from 1-13 written on into a stack. Then every player, numbered from 1-n in clockwise order, takes turn to draw 5 cards from the top of the stack. Every player draws 5 cards in a single time, and then his next player draws cards.

After all the players finish drawing their cards, player 1 will play exactly one card. For simplicity, player 1 will only play the minimum card in his hand, the order of cards is defined as follows: 3<4<5<6<7<8<9<10<11<12<13<1<2.

After player 1’s turn, players from 2-n take their turns in clockwise order, For each player, he should play the card which is exactly the next one of the card played by previous player according the order above. For example, if the previous player played card 4, the current player must play card 5, not card 6,7 or any other card (except card 2).

Card 2 can be played at anyone’s turn as long as his previous player didn’t play card 2. If a player has a card can be played in his hand, he will always play it in his turn. If he can play both 2 and the next card in his turn, he will choose to play the next card first.

If a player can’t play any card, he has to pass his turn and do nothing. If all the players can’t play any card and pass their turns after player X’s turn, all the players from player X should draw one card from the card stack in clockwise order (include player X). After that, player X will play the minimum card in his hand, and the game goes on.

Once there is no card in the stack, skip all the chance for drawing cards, which means if a player need to draw card according the rules above, he will simply ignore this rule and do nothing. But it’s guaranteed that every player will have at least one card in hand before player 1’s first turn.

If one player has no card in his hand at anytime, he will become the winner, and the game ends. Other players should calculate their penalties. The penalty of a player is defined as the sum of numbers written on the cards in his hand.

Now you have known the information about a round of GDY, please find out the result of this round.

Input

There are multiple test cases in the input data.

The first line contains a integer T: number of test cases. $T \leq 50$

For each test case, the first line contain 2 integers n, m, representing the number of players, the number of cards in the original stack. *2≤n≤200, m≤20000.*

The next line contains m integers separated by a blank, representing the original stack. The leftmost one is the top of the stack and the rightmost is the bottom.

For all the test cases, it’s guaranteed that the sum of m doesn’t exceed 4×10^5.

Output

For each test case, print “Case #x:”(without quotes) in the first line, where x is the test case number. Then, print n lines representing the result of each player. For the i-th line, if player i wins, print a string” Winner” (without quotes) at this line. Otherwise, print a integer, the penalty of the i-th player.

Sample Input

```
2
2 10
3 5 7 9 11 4 6 8 10 12
3 15
4 5 6 7 8 9 10 11 12 13 2 2 2 2 2
```

Sample Output

```
Case #1:
Winner
12
Case #2:
26
55
Winner
```

Hint

No response

D-Jerome’s House

Time Limit: 2000/1000 MS (Java/Others)   Memory Limit: 65536K

Problem Description

Jerome just got several large houses as his 18th birthday gifts. To decorate one of these houses, he intends to place several carpets in his house. The house is so large that he can’t find a carpet which can cover the whole house. He’ll book three identical cycle carpets with the radius r. He is unwilling to fold his carpet. So the carpets can’t exceed the house. If the center coordinates of the three carpets are (x1,y1),(x2,y2),(x3,y3).w1=x1*(y2-y3) is the value of the first carpet. w2=x2*(y3-y1) is the second one’s value. And w3=x3*(y1-y2) is the third one’s. He will pay |w1+w2+w3| for these carpets. Jerome is so rich that he wants to maximize his cost. You can just consider this house as a CONVEX polygon.

Input

One integer T ($0<T\leq 30$) in the first line. Indicate the number of the test cases.

In every case:

Two integers in the first line n, r ($3 \leq n \leq 1000, 0 < r < 1000$). Indicate the number of vertexes of the polygon and the radius mentioned above.

n pairs of integers <x, y>($0 \leq |x|,|y| \leq 50000$) in next n lines. Indicate the vertexes of the polygon.

Coordinates of all vertexes are different, and adjacent walls of the room are not collinear. The vertexes are listed in clockwise order.

Output

T lines, output the maximum cost Jerome can pay in each line.

The input data guarantees that the cost is more than 0.

Let the maximum cost is A, and your answer’s cost is B. Your answer will be considered as correct if and only if $|(\mathrm{A}-\mathrm{B})| \leq 10^{-4}$

Sample Input

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

Sample Output

17.7154
1.0000

E-AC Challenge

Time Limit: 2000/1000 MS (Java/Others)   Memory Limit: 128536K

Problem Description

Dlsj is competing in a contest with n (0 < n <= 20) problems. And he knows the answer of all of these problems.

However, he can submit i-th problem if and only if he has submitted (and passed, of course) s_i problems, the $p_{i,1}$-th, $p_{i,2}$-th, …, $p_{i,si}$-th problem before.($0 < p_{i,j} \le n, 0 < j \le s_i, 0 < i \le n$) After the submit of a problem, he has to wait for one minute, or cooling down time to submit another problem. As soon as the cooling down phase ended, he will submit his solution (and get "Accepted" of course) for the next problem he selected to solve or he will say that the contest is too easy and leave the arena.

"I wonder if I can leave the contest arena when the problems are too easy for me."
"No problem."
—— *CCF NOI Problem set*

If he submits and passes the i-th problem on t-th minute(or the t-th problem he solve is problem i), he can get $t \times a_i + b_i$ points. ($|a_i|, |b_i| \le 10^9$)

Your task is to calculate the maximum number of points he can get in the contest.

Input

The first line of input contains an integer, n, which is the number of problems.
Then follows n lines, the i-th line contains $s_i + 3$ integers, $a_i, b_i, s_i, p_1, p_2, \dots, p_{s_i}$ as described in the description above.

Output

Output one line with one integer, the maximum number of points he can get in the contest.

Sample Input

5
5 6 0
4 5 1 1
3 4 1 2
2 3 1 3

1 2 1 4

Sample Output

55

Sample Input

1
-100 0 0

Sample Output

0

Hint

In the first sample.
On the first minute, Dlsj submitted the first problem, and get $1 \times 5 + 6 = 11$ points.
On the second minute, Dlsj submitted the second problem, and get $2 \times 4 + 5 = 13$ points.
On the third minute, Dlsj submitted the third problem, and get $3 \times 3 + 4 = 13$ points.
On the forth minute, Dlsj submitted the forth problem, and get $4 \times 2 + 3 = 11$ points.
On the fifth minute, Dlsj submitted the fifth problem, and get $5 \times 1 + 2 = 7$ points.
So he can get $11 + 13 + 13 + 11 + 7 = 55$ points in total.

In the second sample, you should note that he doesn’t have to solve all the problems.

F-An Easy Problem On The Trees

Time Limit: 2000/1000 MS (Java/Others) Memory Limit: 262144K

Problem Description

In this problem you will be given a tree with N nodes and N - 1 bidirectional edges. Now you should handle three kinds of operations.

1. Given two integers x, y, you should make a new edge between these two node x and y if they belong to different component before.

2. Given two integers x, y, if they belong to the same component, you should find the tree in the tree set which contains node x, and you should make the node x be the root of this tree, and then you should cut the edge between node y and its father.

3. Given one integer x. It means that if there is one point on the node x initially, and then it moves to all adjacent nodes with the same probability, it will keep moving like this until it goes back to the node x. You should output the expected number of edges that it will go through.

Input

The first line contains two integers N and M which means there are N nodes in the tree and M operations you need to handle. In the next N – 1 lines, each line contains two integers x and y, representing an edge between x and y.

The next M lines start with an integer t (t in {1, 2, 3}), representing the kind of this operation.

It's guaranteed that 1 <= N, M <= 100000, 1 <= x, y <= N and the N nodes form a tree initially.

Output

For operation 1 and 2, if the operation is illegal, you should output -1. For operation 3, if there is no other node in the component which node x belongs to, you should output 0. Otherwise you should output the the expected number of edges modulo 998244353 in the format $p*q^{-1}$. It's guaranteed that the the answer is always p / q (p and q are two coprime integers).

Sample Input

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

Sample Output

```
-1
2
0
```

G-Lpl and Energy-saving Lamps

Time Limit: 2000/1000 MS (Java/Others)   Memory Limit: 65536K

Problem Description

During tea-drinking, princess, amongst other things, asked why has such a good-natured and cute Dragon imprisoned Lpl in the Castle? Dragon smiled enigmatically and answered that it is a big secret. After a pause, Dragon added:

— We have a contract. A rental agreement. He always works all day long. He likes silence. Besides that, there are many more advantages of living here in the Castle. Say, it is easy to justify a missed call: a phone ring can't reach the other side of the Castle from where the phone has been left. So, the imprisonment is just a tale. Actually, he thinks about everything. He is smart. For instance, he started replacing incandescent lamps with energy-saving lamps in the whole Castle...

Lpl chose a model of energy-saving lamps and started the replacement as described below. He numbered all rooms in the Castle and counted how many lamps in each room he needs to replace.

At the beginning of each month, Lpl buys m energy-saving lamps and replaces lamps in rooms according to his list. He starts from the first room in his list. If the lamps in this room are not replaced yet and Lpl has enough energy-saving lamps to replace all lamps, then he replaces all ones and takes the room out from the list. Otherwise, he 'll just skip it and check the next room in his list. This process repeats until he has no energy-saving lamps or he has checked all rooms in his list. If he still has some energy-saving lamps after he has checked all rooms in his list, he'll save the rest of energy-saving lamps for the next month.

As soon as all the work is done, he ceases buying new lamps. They are very high quality and have a very long-life cycle.

Your task is for a given number of month and descriptions of rooms to compute in how many rooms the old lamps will be replaced with energy-saving ones and how many energy-saving lamps will remain by the end of each month.

Input

Each input will consist of a single test case.

The first line contains integers n and m ($1 \le n \le 100000$, $1 \le m \le 100$) — the number of rooms in the Castle and the number of energy-saving lamps, which Lpl buys monthly.

The second line contains n integers $k_1, k_2, ..., k_n$ ($1 \le k_j \le 10000$, $j = 1, 2, ..., n$) — the number of lamps in the rooms of the Castle. The number in position j is the number of lamps in j-th room. Room numbers are given in accordance with Lpl's list.

The third line contains one integer q ($1 \le q \le 100000$) — the number of queries.

The fourth line contains q integers $d_1, d_2, ..., d_q$ ($1 \le d_p \le 100000$, $p = 1, 2, ..., q$) — numbers of months, in which queries are formed.

Months are numbered starting with 1; at the beginning of the first month Lpl buys the first m energy-saving lamps.

Output

Print q lines.

Line p contains two integers — the number of rooms, in which all old lamps are replaced already, and the number of remaining energy-saving lamps by the end of d_p month.

Sample Input

```
5 4
3 10 5 2 7
10
5 1 4 8 7 2 3 6 4 7
```

Sample Output

```
4 0
1 1
3 6
5 1
5 1
2 0
3 2
4 4
3 6
5 1
```

Hint

Explanation for the sample:

In the first month, he bought 4 energy-saving lamps and he replaced the first room in his list and remove it. And then he had 1 energy-saving lamps and skipped all rooms next. So, the answer for the first month is 1,1------1 room's lamps were replaced already, 1 energy-saving lamp remain.

H-Set

Time Limit: 3000/1500 MS (Java/Others)   Memory Limit: 524288K

Problem Description

Shinku is very interested in the set. One day, she got n sets, and the i-th number a_i is in the i-th set. But she doesn't think it is interesting enough, so she applies m magic to these sets. There are three kinds of magic:

1 u v: If the u-th and v-th numbers are not in one set, then the Shinku's magic will merge the set containing the u-th number and the set containing the v-th number.

2 u: Shinku's magic adds 1 to each number in the set containing the u-th number.

3 u k x: Shinku can immediately know how many numbers t in the set containing the u-th number satisfy $t \equiv x \pmod{2^k}$ $(0 \le k \le 30, 0 \le x < 2^k)$.

But unfortunately, for some reason the type 3 magic fails. So Shinku wants you to tell her the answer after every type 3 magic.

Note that there can be multiple numbers with the same value in one set, that is, numbers with the same value will not disappear when merged.

Input

The first line contains two integers n, m$(1 \le n, m \le 6 \times 10^5)$,the number of initial sets and the number of the magic.

The second line contains n integers. The i-th number $a_i (0 \le a_i \le 10^9)$is the number in the i-th set initially.

The next m lines describe the sequence of magic. The i-th line describes the i-th magic. Each magic is a magic as described above.

Output

For each type 3 magic, output the answer you are asked to calculate.

Sample Input

```
3 5
2 3 4
1 1 3
3 3 1 0
2 2
1 2 3
3 3 1 0
```

Sample Output

```
2
3
```

Hint

After the first operation, the numbers are 2,3,4, sets are {2,4} {3}

For the second operation, the third number is in {2,4}, $2 \equiv 0 (\mathrm{mod}\ 2^1), 4 \equiv 0 (\mathrm{mod}\ 2^1)$,so the answer is 2.

After the third operation, the numbers are 2,4,4, sets are {2,4} {4}

After the forth operation, the numbers are 2,4,4, sets are {2,4,4}

For the fifth operation, ,the third number is in {2,4,4}, $2 \equiv 0 (\mathrm{mod}\ 2^1), 4 \equiv 0 (\mathrm{mod}\ 2^1), 4 \equiv 0 (\mathrm{mod}\ 2^1)$,so the answer is 3.

I-Skr

Time Limit: 2000/1000 MS (Java/Others)   Memory Limit: 256000K

Problem Description

A number is skr, if and only if it's unchanged after being reversed. For example, "12321", "11" and "1" are skr numbers, but "123", "221" are not. FYW has a string of numbers, each substring can present a number, he wants to know the sum of distinct skr number in the string. FYW are not good at math, so he asks you for help.

Input

The only line contains the string of numbers S.

It is guaranteed that $1 \leqslant S[i] \leqslant 9$, the length of S is less than 2000000.

Output

Print the answer modulo 1000000007.

Sample Input

111111

Sample Output

123456

Sample Input

1121

Sample Output

135

J-Sum

Time Limit: 2000/1000 MS (Java/Others)   Memory Limit: 512000K

Problem Description

A square-free integer is an integer which is indivisible by any square number except 1. For example, $6 = 2 \cdot 3$ is square-free, but $12 = 2^2 \cdot 3$ is not, because 2^2 is a square number. Some integers could be decomposed into product of two square-free integers, there may be more than one decomposition ways. For example, $6 = 1 \cdot 6=6 \cdot 1=2 \cdot 3=3 \cdot 2$, $n=ab$ and $n=ba$ are considered different if $a \neq b$. $f(n)$ is the number of decomposition ways that $n=ab$ such that a and b are square-free integers. The problem is calculating $\sum_{i=1}^{n} f(i)$.

Input

The first line contains an integer $T(T \leqslant 20)$, denoting the number of test cases. For each test case, there first line has a integer $n(n \leqslant 2 \cdot 10^7)$.

Output

For each test case, print the answer $\sum_{i=1}^{n} f(i)$.

Sample Input

2
5
8

Sample Output

8
14

Hint

$\sum_{i=1}^{8} f(i) = f(1) + \cdots + f(8) = 1 + 2 + 2 + 1 + 2 + 4 + 2 + 0 = 14$

K-The Great Nim Game

Time Limit: 2500/1500 MS (Java/Others)   Memory Limit: 131072K

Problem Description

Nim is a famous game as you know. Nim is a 2-player game featuring several piles of stones. Players alternate turns, and on his/her turn, a player's move consists of removing one or more stones from any single pile. Play ends when all the stones have been removed. The first player who can't remove is declared as the loser.

Now you want to play the Great Nim Game. In the other words, you want to choose several (0~N) pile(s) from N piles of stones. You want know how many choices you have making sure that the first player must win. They both try their best (optimal strategy) to win through the game.

Input

The first line contains two numbers N x_1, denoting the number of piles and the number of stones in the first pile.

The second line contains five integers a, b, c, d, e.

The third line contains one integer k, denoting a function $f(x) = (ax^4+bx^3+cx^2+dx^1+e-1)\%k+1$. With these, you can figure out the number of stones in the i-th pile $x_i = f(x_{i-1})$ $(1<i\leqslant N)$

It is guaranteed that $1<N<10^{10000000}$, $0<x_1\leqslant k$, $0\leqslant a, b, c, d, e<2^{12}$, $a+ b+ c+ d+ e>0$, $0<k<2^{12}$

Output

Print the number of solutions making sure the first player must win. The answer may be very large, so you should output it mod 1e9+7 (% 1000000007).

Sample Input #1

5 1
0 0 0 1 1
16

Sample Output #1

28

Sample Input #2

100000000000000000000 1
0 0 0 1 1
4095

Sample Output #2

394326889

Sample Input #3

100000000000000000000 1
1 0 0 1 1
4095

Sample Output #3

933180537

Hint

In the first sample, there are 1, 2, 3, 4, 5 stones in the 1-st, 2-nd, 3th, 4th, 5th pile. You can figure out there are exactly 4 choosing ({1,2,3} {1,4,5} {2,3,4,5} {}(empty, you choose zero pile)) ways that make first-hand player must lose, so the answer is 2^5-4=28.

If x is in range (1⩽x⩽k), f(x) must be in range (1⩽f(x)⩽k), too.

L-Magical Girl Haze

Time Limit: 2000/1000 MS (Java/Others) Memory Limit: 256MB

Problem Description

There are N cities in the country, and M directional roads from u to v(1<=u, v<=n).Every road has a distance ci. Haze is a Magical Girl that lives in City 1, she can choose no more than K roads and make their distances become 0. Now she wants to go to City N, please help her calculate the minimum distance.

Input

The first line has one integer T(1<=T<=5), then following T cases.

For each test case, the first line has three integers N, M and K. Then the following M lines each line has three integers, describe a road, Ui, Vi, Ci. There might be multiple edges between u and v.

It is guaranteed that N<=100000, M<=200000, K<=10, 0<=c[i]<=1e9. There is at least one path between City 1 and City N.

Output

For each test case, print the minimum distance.

Sample Input

```
1
5 6 1
1 2 2
1 3 4
2 4 3
3 4 1
3 5 6
4 5 2
```

Sample Output

```
3
```