2024 Florida International University
High School Programming Competition

Problems - Standard Division

	Problem Name	Filename
A.	Alphabetical Anomalies!	alphabet
B.	Panther Optimization	panther
C.	Server Assignment	server
D.	A Day Off	dayoff
E.	Dizzying Distributions	dizzy
F.	Secret Santa	santa
G.	Jelly Fishing	jelly
H.	Unstable Compounds	unstable
I.	Portal Madness	portal
J.	Tree Counting	tree

Call your program file: filename.cpp, filename.java, or filename.py

A. Alphabetical Anomalies!

Aaarush and Bronit have a string of letters s of length n that they enjoy staring at very much; heck, if they didn't have other things to worry about like paying taxes and winning programming competitions, they would probably like to stare at strings of letters all day long! But there's something up with their string of letters: some characters aren't in order! More formally, there may exist pairs of indices, $0 \leq i < j < n$, such that s_j is lexicographically before s_i. Since Aaarush and Bronit aren't the sharpest tools in the shed, they like to keep their string simple, having only the characters `a` and `b` in their string. They would also like you to help them figure out how many characters should be removed from their string such that there are no longer any pairs of characters that are out of order, and that the number of removals made is minimal.

Input

The first and only line of input will contain a single string s of length $n, 1 \leq n \leq 10^5$.

Output

Output the minimum number of removals so that all characters are in order.

Sample Input 1	**Sample Output 1**
`aaaabbb`	`0`

Sample Input 2	**Sample Output 2**
`aababbab`	`2`

Sample Input 3	**Sample Output 3**
`bbaaaaabb`	`2`

Sample Input 4	**Sample Output 4**
`aabaabbab`	`2`

B. Panther Optimization

Geeyawn is a student at the Paul L. Cejas School of Architecture here at FIU, and he has just received his homework from one of his professors: the pathways connecting certain buildings have eroded sufficiently enough such that FIU now needs to do something about it. For the assignment, Geeyawn's professor has given him a list of the buildings of concern, as well as a list of pairs of these buildings for which the pathway connecting them needs fixing, along with the cost of fixing each pathway. In giving Geeyawn this assignment, his professor has noted the following:

- Not all pathways need to be fixed. The professor wants the students to choose a subset of the pathways such that the total cost of fixing the chosen pathways is *minimized*, and that it would be possible to reach every building from every other building using the fixed pathways.
- The professor has guaranteed the existence of a set of pathways to satisfy the above conditions. In fact, it may be that there are *multiple* valid answers, so the the students may pick one of them arbitrarily. The pathways to be fixed may be output in any order the students chooses.
- Connecting any two buildings, there is *at most* one pathway.

Geeyawn's ability in architecture is rather poor, and so he has sought your guidance in completing this homework assignment.

Input

The first line of the input will contain two space-separated integers n and m, where $1 \le n \le 500000$ and $n-1 \le m \le \min\{500000, \frac{n(n-1)}{2}\}$. Here n is the number of buildings to be considered, and m is the total number of pathways that need fixing.

The next n lines of the input will each contain a single string $s, 1 \le |s| \le 10$, containing uppercase English letters or digits, which represent the buildings of concern in this ordeal.

The next m lines of the input will each contain pairs of building names a and b, as well as a positive integer $c, 1 \le c \le 5000$, all space-separated. Thus, the cost of fixing the pathway connecting the buildings a and b will be c dollars.

Output

The first line of output should contain a single integer u, the total cost of fixing the optimal set of pathways. The following lines, one for each pathway in the optimal set of pathways to be fixed, should contain two strings, space-separated, denoting the pair of buildings connected by the pathways chosen to be fixed.

Sample Input 1

```
4 4
A
B
C
D
A B 1
B D 3
B C 2
C D 0
```

Sample Output 1

```
3
A B
B C
C D
```

Sample Input 2

```
8 8
A
B
C
D
E
F
G
H
A B 3
A D 3
A F 4
B C 1
B H 1
C G 2
E F 1
D E 1
```

Sample Output 2

```
12
B C
B H
D E
E F
C G
A B
A D
```

C. Server Assignment

Ricky has been setting up this year's high school programming competition. He has been working on setting up the judging servers. During the contest, contestants will a series of submissions which will be automatically judged by a computer. A pool of computers have been set aside to do the judging, and submissions will be assigned to computers from this pool to judge. Each submission has two pieces of information, the submission time and the time it takes to judge that submission. This being a tedious process, we would like to automate it. Since Ricky has been busy, he doesn't have time to implement this, but he does know how computer assignment should be handled. Initially, there will be n computers with IDs 1 through n, when a contestant makes a submission, if there are multiple computers available, the computer that has spent the least total amount of time judging submissions will be assigned to judge that submission; if two or more computers are tied for the least amount of time spent judging, then the one with the smallest ID will be picked. If no computers are available, then the submission will be placed in a queue to wait for the first available computer. Multiple submissions may be in the queue at any given time. If multiple submissions arrive at the same time, they will be ordered by increasing time needed to judge them.

Input

The first line of the input will contain two integers m ($1 \leq m \leq 10^5$) and n ($1 \leq n \leq 10^5$), the number of submissions and the numbers of computers, respectively. The following m lines will each contain two integers t ($1 \leq t \leq 10^9$) and v ($1 \leq v \leq 10^9$), where t is the time of the submission and v is the time it takes to judge that submission.

Output

Output n space-separated integers: the total time that each of the n computers spent on judging problems, going in ascending order of ID.

Sample Input 1

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

Sample Output 1

```
17 7
```

Initially two submission arrive at $t = 1$ the one that takes 4 seconds to process is given to computer 1. The second one is given to computer 2.

Sample Input 2

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

Sample Output 2

```
14 8
```

D. A Day Off

Aaarush and Bronit are on their day off, and they're up to their usual shenanigans and antics as they're making their way to the Green Library so they can acquire some of that sweet, sweet Starbucks. On their way to the Green Library, however, they spotted some stones of varying sizes arranged in a single file right beside the lake in front of the Green Library. Was this the work of mother nature, or did some other student have a hand in this? Since thinking is too hard for Aaarush and Bronit (evident from their scores on their Programming 1 finals last semester), they promptly dismissed such thoughts, and started playing with the stones. They learned that each stone also had an associated value. They devised the following game with the stones: In a single turn, a player must take at least one and no more than three stones from the *beginning* of the row of stones. Once they do this, the sum of the values of the stones that player took is added to their score. They repeat this until no stones remain, which is when the game ends. Whichever player ends with a higher score wins, unless both of their scores are equal, in which case they call it a tie. Aaarush will always be playing first because he *just barely* beat Bronit's score on their Programming 1 final. Since Aaarush and Bronit don't know how to play this game optimally, they would like you, an expert game theorist, to tell them which one of them would win this game under the assumption that they both play optimally.

Input

The first line of the input will contain a single integer $1 \leq n \leq 50000$, the number of stones they found along the lake. The second line of the input will contain a list $a_1, a_2, \ldots, a_n$ of the values of stones listed in a space-separated manner. Note that $1 \leq |a_i| \leq 1000$.

Output

Output the player who will end up with the greater sum and will therefore win, assuming they play optimally. Output `Aaarush` if Aaarush will win, output `Bronit` if Bronit will win, and output `Tie` if the game will end in a tie.

Sample Input 1

```
4
1 2 3 7
```

Sample Output 1

```
Bronit
```

For the above case, no matter how many stones Aaarush takes, Bronit will always be able to take *at least* the last stone, which makes him the winner, as that stone has a sum greater than the first three stones.

Sample Input 2

```
4
1 2 3 6
```

Sample Output 2

```
Tie
```

For the above case, Aaarush can force a tie by taking all three of the first stones in the row (any other move would *not* be optimal for Aaarush).

Sample Input 3

```
5
1 2 -5 1 3
```

Sample Output 3

```
Aaarush
```

For the above case, Aaarush's optimal move would be to take the first two stones. Then, Bronit makes his optimal move of taking the last three stones, but because the third stone was just so ugly (and he must take *at least* this stone), Aaarush ends up being victorious.

E. Dizzying Distributions

David works at a factory that distributes letters from a pool of letters that originated somewhere else in the factory. As an employee at the factory, it is David's job to put these letters in a certain arrangement before they're ready to be shipped. However, David wasn't paying attention when his boss was explaining how these letters were intended to be arranged, as he was too busy trying to understand a new algorithm he had learned a few hours prior for a programming competition he had coming up in a few weeks. Consequently, David instead does as he pleases with the letters that come his way, but he notices that sometimes, after he's finished arranging the letters, he gets dizzy when he stares at them, but thankfully, this only happens when there are clusters of two or more consecutive letters in the final arrangement. Given the pool of letters that are coming David's way, he requires your assistance in ordering the letters in the pool in a fashion such that he will not get dizzy when he stares at them.

Input

The first and only line of the input is a single string s of length at most 100000 sent to David.

Output

If possible, print a rearrangement of the letters in the input string such that David will not get dizzy when he stares at them. If there are multiple such arrangements, print any one of them. If it is not possible, print "`:(`" (without the quotes).

Sample Input 1

```
aab
```

Sample Output 1

```
aba
```

Sample Input 2

```
aaab
```

Sample Output 2

```
:(
```

Sample Input 3

```
vvvlo
```

Sample Output 3

```
vlvov
```

F. Secret Santa

Your class is doing secret Santa this year. Everyone will come to class with a present and a person they will be giving the present to. Simple enough, right? You, being the curious person that you are, think about a problem relating to this year's secret Santa: what if just presenting gifts once, you swap presents many times? More formally, you are given an array a of length n, where a_i represents person i will give a gift to a_i. You will give each other gifts until everyone ends up with the gift they came with. It is guaranteed each person will not receive a gift from more than one person, and therefore a will be a permutation of the first n natural numbers.

Input

The first line of input will contain one integer n ($1 \leq n \leq 10^5$), the length of the array representing whom each person will give their gift to. The next line of input will contain the array a, each integer separated by a single space.

Output

Output the number of rounds for the gifts to return to their original owners.

Sample Input 1

```
3
2 3 1
```

Sample Output 1

```
3
```

Explanation: Initially person 1 will come with gift a, person 2 will come with gift b, and person 3 will come with gift c.
After one round person 1 will have gift c, person 2 will have gift a, person 3 will have gift b.
After round two person 1 will have gift b, person 2 will have gift c, person 3 will have gift a.
After round three person 1 will have gift a, person 2 will have b, and person 3 will have c.
So after three rounds everyone ends up with the gift they brought.

Sample Input 2

```
4
2 1 4 3
```

Sample Output 2

```
2
```

G. Jelly Fishing

Patrick and Spongebob enjoy jelly fishing. They like to go out every weekend and catch as many jellyfish with their nets as they can. Currently, all the jellyfish are on the ground and they would like to position their net to maximize the number of jellyfish that lie inside the net in one swoop. The jellyfish are quite small, so they can just be modeled as points in a 2 dimensional grid. Any jellyfish that lies on the edge of the net is still considered caught.

Input

The first number in the input will be an integer n ($1 \leq n \leq 100$) that denotes the number of test cases which follow.
A blank line comes at the beginning of each test case. Next we have a line containing an integer m ($1 \leq m \leq 32$), representing the number of jellyfish on the ground, and a real number d ($0 \leq d \leq 200$) representing the diameter of the net that SpongeBob and Patrick are using. Each of the following lines will specify the location of a jellyfish in the form of real coordinates x ($-100 \leq x \leq 100$) and y ($-100 \leq y \leq 100$).

Output

For each test case, you are to print the maximum number of jellyfish that can be caught under the net, if optimally positioned. You may assume that the answer would not change if the diameter of the net is increased by at most 10^{-5}.

Sample Input 1

```
2

4 1.5
1.0 3.75
3.0 1.0
1.0 2.25
1.5 3.0

8 3.0
-1.0 3.0
-1.0 2.0
-2.0 1.0
0.0 1.0
1.0 0.0
1.0 -1.0
2.0 -2.0
3.0 -1.0
```

Sample Output 1

```
3
4
```

H. Unstable Compounds

Your sibling is a chemistry nerd and loves creating new compounds by mixing chemicals. He has all his n chemicals lined up on the table. Each chemical has a value associated with it, which are given by $A_1, A_2, \ldots, A_n$. When two chemicals in positions i and j are combined, they create a compound with value $A_i \oplus A_j$. Chemicals placed close to each other are highly reactive. Your sibling plans to combine pairs of chemicals but wishes to avoid combining two reactive ones. You are also given a fixed integer k. You have been informed that two chemicals i and j are reactive if $|i - j| < k$. Your sibling would like to find the pair of chemicals, which minimizes the value of the resulting compound when combined. Help your sibling by writing a program that finds the two chemicals that minimize this value, but which are at distance k or greater. *Note $\oplus$ of two integers is their "exclusive or" when represented as binary numbers.

Input

The first line of the input contains two integers n ($3 \leq n \leq 10^5$), the length of the array, and k ($1 \leq k \leq n-1$), the smallest distance two chemicals can be from each other while being considered a stable pair. The following line contains the array A as a sequence of space-separated integers, where $1 \leq A_i \leq 10^9$.

Output

There should be one line of output which is the minimum value of a compound obtained by combining two chemicals at a distance k or greater.

Sample Input 1

```
5 2
1 1 1 2 2
```

Sample Output 1

```
0
```

I. Portal Madness

Ricky (the smartest mammal in the galaxy) and his grandchild Marty have been exploring the galaxy in search of mega seeds. Unfortunately, at some point in their adventure, Marty got lost along the way at one of the planets they visited. Ricky is now tasked with finding him, or his daughter, Betsy, will never talk to him again. There are n planets numbered from 0 to $n-1$, and Ricky is currently on Earth, which is planet 0. Ricky is able to create portals to travel around planets, but creating portals takes time. There are two types of portals Ricky can open: normal portals, which connect two planets u, v and take d units of time to open up; and special portals - these special portals can be opened up at certain planets and open up a range of planets for a one time use. More formally, these portals are described by (v, L, R, d) where v is the planet you need to be to open up the portal and that portal allows you to access planets in the range $[L, R]$ and takes d units of time to open up. Ricky also hates waiting consecutive periods of time, so he wants to minimize the maximum time spent on a single portal when reaching some planet. If Ricky opens up k portals $A_1, A_2, ..., A_k$ to reach planet i, then you want to minimize $\max(A_1.d, A_2.d, ..., A_k.d)$.

Input: The first line of the input will contain two integers $1 \leq n \leq 10^5$, the total number of planets visited, $1 \leq m \leq 10^5$ is the number of portals that can be created. The next m lines contain t, u, l, r, d if t = 1, d or t, u, v, d if t = 2 where $1 \leq t \leq 2$, $0 \leq u, v, l, r < n$ and $0 \leq d \leq 10^9$.

Output: n integers on a line; i-th integer is cost to travel from planet 0 to i-th planet.

-1 if not possible

Sample Input 1

```
5 5
2 0 4 10
2 1 4 1
2 3 4 2
1 0 1 2 4
2 0 3 1
```

Sample Output 1

0 4 4 1 ~~4~~ 2

Explanation: The shortest path from planet 0 to 0 is obvious. To reach planet 1 you can use the special portal to travel to 1 at cost 4. To reach planet 2 is the same as planet 1 also at a cost of 4. Planet 3 has a portal of cost 1 with a single portal. For planet 4 you will need two portals first go to planet 1 with the special portal and then use the portal of cost 1. Therefore the distance is max(1, 4) = 4

J. Tree Counting

Given a positive integer n, count the number of unlabelled, rooted binary trees whose diameters are n.

Input

The first line of input contains one integer, n ($1 \leq n \leq 100$).

Output

The first and only line of output contains one integer, the answer to this problem. As it may be large, give it modulo $10^9 + 7$.

Sample Input 1

```
1
```

Sample Output 1

```
2
```

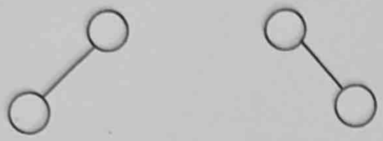

Figure 1: Above are all rooted binary trees with a diameter of 1.

Sample Input 2

```
2
```

Sample Output 2

```
7
```

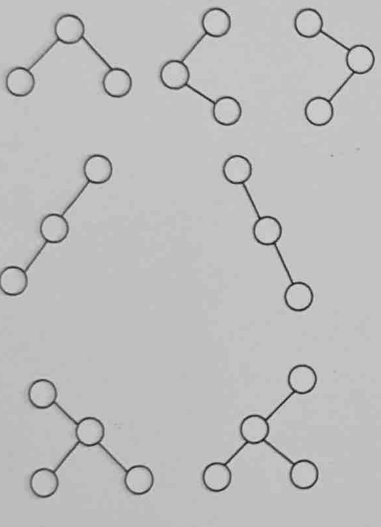

Figure 2: Above are all rooted binary trees with a diameter of 2.