CodeNection 2023 Final Round Problems

Competition Team of CodeNection 2023

Contents

Codey and CodeNection 2

Problem Statement

Codey worked really hard and its perseverance has landed it in the CodeNection Finals! Codey wishes to express its love for CodeNection, but this time, it wants you to put a clever twist. Reverse the phrase `I LOVE CODENECTION` n times, and print the **final string**.

Input Format

The first line contains an integer n, which represents the number of times Codey wants to reverse the string `I LOVE CODENECTION`.

Constraints

$0 \leq n \leq 10^{18}$

Output Format

Output the string `I LOVE CODENECTION` reversed n times.

Sample Test Case 1

3	NOITCENEDOC EVOL I

`I LOVE CODENECTION` is reversed 3 times.

`NOITCENEDOC EVOL I` $\rightarrow$ `I LOVE CODENECTION` $\rightarrow$ `NOITCENEDOC EVOL I`

Codey and Connection

Problem Statement

Codey is a huge fan of CodeNection, and one day, while enjoying a CodeNection-themed article he noticed that the word 'Connection' closely resembles 'CodeNection'. Excited by this idea, Codey prepared a sequence of n **uppercase** alphabetical letters, all lined up from left to right. Codey decided to connect these letters following a specific rule:

- Let the index of the previously connected letter be i, and pick an integer j where $i < j \leq n$. If no letter was connected previously, Codey could pick any i.
- Connect i to j.

Codey is now curious to know if it's possible to **connect** these letters in a way that spells out the word `CODENECTION` from **left to right**. Can you help Codey figure it out?

Input Format

The first line contains an integer t, which represents the amount of test cases for the problem.

The following provides the description of each test case:

- The first line contains an integer n, which represents the number of letters in the sequence of uppercase letters.
- The second line contains the sequence of n uppercase letters.

Constraints

- $1 \leq t \leq 100$
- $11 \leq n \leq 10^3$

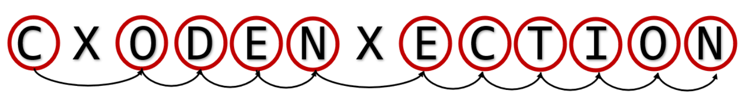
Output Format

Output `YES` for each test case if its possible to spell out `CODENECTION` otherwise output `NO`.

Sample Test Case 1

2 13 C X O D E N X E C T I O N 13 L O V E F R O M C O D E Y	YES NO

In the first test case, it is possible to connect the letters, and spell out `CODENECTION`.



whereas for the second test case, it is not possible to connect in such a way that spells out `CODENECTION`.

Codey and Schedule

Problem Statement

Codey is the chairman of a competition with n teams and the game cannot start without a schedule. Due to time constraints, Codey wants to arrange a schedule with a minimal number of rounds so that each of the n team will play against **two different teams**. Help Codey arrange the schedule!

Output m, which represents the number of rounds, and m pairs of integers a_i and b_i ($a_i \neq b_i$), which represent the i-th match up in the schedule. **It is guaranteed that the answer exists**.

Note that this problem uses a custom checker, so make sure your program compiles correctly and prints the output according to the format before submitting.

Input Format

The first line contains an integer n, which represents the number of teams.

Constraints

$3 \leq n \leq 10^5$

Output Format

Output an integer m, which represents the number of rounds, for the first line.

In the following m lines, output pairs of integers a_i and b_i, where $a_i \neq b_i$, representing the i-th matchup in the schedule.

If there are multiple answers, you may output any of them.

It is guaranteed that the answer exists.

Sample Test Case 1

5	5 3 2 3 4 5 1 5 4 2 1

There are 5 rounds in total.

In the first round, team 3 will play against team 2.
In the second round, team 3 will play against team 4.
In the third round, team 5 will play against team 1.
In the forth round, team 5 will play against team 4.
In the fifth round, team 2 will play against team 1.

Codey and School Supplies

Problem Statement

As a new semester begins, Codey, a dilligent student, steps into a local stationery shop, ready to assemble a complete set of school supplies for its upcoming academic journey. Codey requires three essential school supplies: an eraser denoted as `A`, a pencil denoted as `B`, and a ruler denoted as `C`. The stationery shop offers n sets of school supplies, each with its own price, a_i and contents, b_i.

Codey can purchase multiple sets of school supplies, and its goal is to find the **minimum** total cost to obtain **at least** one eraser, one pencil, and one ruler. Can you assist Codey in finding the minimum total cost to acquire at least one of each of the required school supplies (eraser, pencil, and ruler) from the stationery shop?

Input Format

The first line contains an integer n, which represents the number of sets of school supplies offered in the stationery shop.

The second lines contains an integer a_i and a string b_i, where a_i represents the price of the i-th set of school supplies, and b_i represents the contents of the school supplies set.

Constraints

- $1 \leq n \leq 10^3$
- $0 \leq a_i \leq 10^3$
- String b_i consists of 1 to 3 characters, with the only valid characters being `A`, `B` and `C`. The arrangement of letters within string b_i is random.

Output Format

Output the minimum cost that Codey needs to obtain at least an eraser, a pencil, and a ruler. If there is no way to obtain at least one of each of the three items, output -1.

Sample Test Case 1

3 10 CBA 5 BC 3 A	8

Codey can acquire the 2nd, and 3rd school supplies sets by spending $5 + 3 = 8$, thus obtaining all three items. This is a better choice compared to obtaining the 1st set.

Sample Test Case 2

2 5 B 3 C	-1

Codey is unable to obtain all three items, since there is no school supplies set containing `A` — an eraser.

Codey and Zombies

Problem Statement

Codey is on its way back home, as usual. To get there, it has to traverse a field that can be represented as an $n \times m$ grid. However, today is different because the field is inhabited by zombies! Codey is aware that the only way to eliminate the zombies is by setting them on fire.

So, Codey has devised an ingenious plan for this task, involving the use of bombs. Codey can place a bomb at a cell $(i,\ j)$, and doing so requires a total of $a_i + b_j$ units of fire powder. Codey is allowed to place **at most** one bomb on each cell $(i,\ j)$. When Codey detonates a bomb in cell $(i,\ j)$, it results in the burning of every cell in row i and every cell in column j. This means that all zombies in the corresponding rows and columns will be killed when Codey detonates the bombs.

What are the **minimum** and **maximum** amounts of fire powder Codey could use to guarantee the elimination of every zombie?

Input Format

The first line contains n and m, where n represents the number of rows and m represents the columns in the 2D grid.

The second line contains n integers, $a_1, a_2, a_3, ..., a_i$, each representing the values fire powder for each row.

The third line contains m integers, $b_1, b_2, b_3, ..., b_j$, each representing the values fire powder for each column.

Constraints

- $1 \leq n, m \leq 10^5$

- $1 \leq a_i, b_i \leq 10^5$

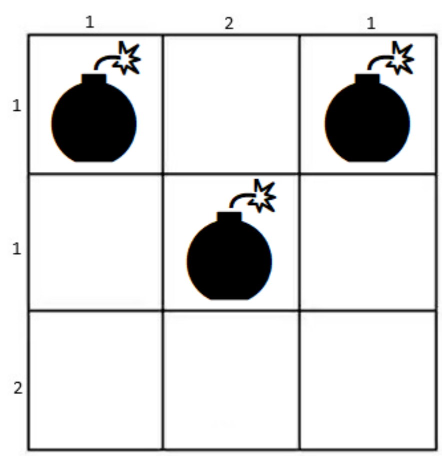
Output Format

Output two integers, maximum and minimum amount of firepower Codey can use.

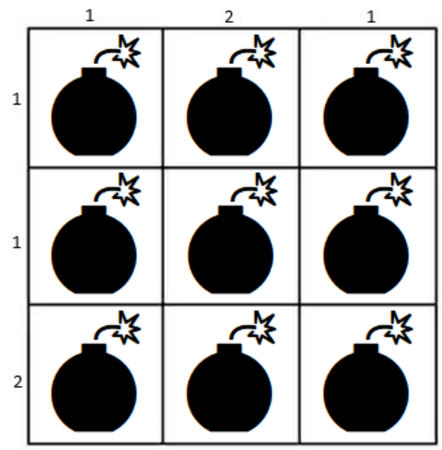
Sample Test Case 1

3 3 1 1 2 1 2 1	7 24

The picture below shows the position of the bombs that can ensure the minimum amount of fire powder used $(1+1)+(1+2)+(1+1)=7$.



And this is the maximum amount of fire power Codey can use to eliminate all the zombies which is 24.


1
2
1
1
1
2

Codey and Sightseeing

Problem Statement

Codey has an adventurous spirit. Whatever catches its eye, it will take a look at. There are n locations, m roads, and k of the locations that have attracted Codey's attention. Each road represents a distance of 1. You will be given an array a of length k, where a_i represents the i-th place of interest.

Due to its curiosity, Codey is eager to find the **shortest distance** it needs to take to reach the closest place of interest from each location. Your task is to find the **sum of distances** from every location to its closest place of interest.

Input Format

The first line contains three integers, n, m and k, where n represents the total number of locations, m represents the number of roads and k represents the number of places of interest.

The second line contains k integers $a_1, a_2, ..., a_k$, each representing the i-th place of interest.

The following m lines contains two integers, u and v, which indicates the road between location u and v.

Constraints

- $1 \leq n \leq 10^5$
- $n - 1 \leq m \leq n \cdot \frac{(n-1)}{2}$
- $1 \leq a_i \leq n$, where all a_i are unique

- It is guaranteed that there is no self loop & multiple edges for each test case

Output Format

Output an integer representing the sum of distances from every location to its closest place of interest.

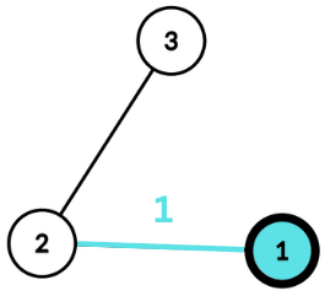
Sample Test Case 1

3 2 1 1 2 3 2 1	3

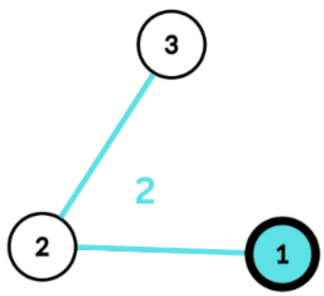
There are a total distance of 3 from each location to the place of interest as follows:

For location 1, the closest place of interest is itself, therefore distance is 0.

For location 2, the closest place of interest is location 1, therefore distance is 1.



For location 3, the closest place of interest is location 1, therefore distance is 2.



Therefore, the sum of distances from each location to its closest place of interest is $0 + 1 + 2 = 3$.

Codey and Apples

Problem Statement

An apple a day keeps the doctor away. Codey is a good friend and would like to keep its friends away from the doctors and keep them healthy.

Codey has n bags, where the i-th bag contains a_i apples, and it has decided to share its apples with its m friends. In order to keep this fair, Codey will only share the apples with its friends if it can choose some bags (at least 1 and possibly all) and evenly divide all the apples in those bags among the m friends. In other words, the total number of apples in the bags Codey has chosen must be divisible equally among the m friends.

Your task is to find the bags that meet this condition. If there are multiple answers, you may print any of them. If it is not possible, output -1.

Note that this problem uses a custom checker, so make sure your program compiles correctly and prints the output according to the format before submitting.

Input Format

The first line of input contains integer n, m, which represent the number of bags and the number of friends, respectively.

The second line of input contains n integers $a_1, a_2, ..., a_n$, each representing the number of apples in the i-th bag.

Constraints

- $1 \leq m \leq n \leq 10^5$
- $1 \leq a_i \leq 10^5$

Output Format

Output the number of bags chosen, k ($1 \leq k \leq n$) in the first line if it meets the condition.

In the next line, output k unique integers, which are the indices of the selected bags for the second line.

If there are multiple answers, you may print any of them. If it is not possible, output -1.

Sample Test Case 1

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

Codey can choose the fourth bag and get 4 apples. Codey can then split one apple to each friend.

Sample Test Case 2

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

Codey can choose the first 4 bags and get $1+1+1+1=4$ apples. Codey can then split one apple to each friend.

Codey and Facto

Problem Statement

Codey is very interested in *facto*, a concept he learned in the mathematics class last week.

If you are given an integer n and an integer m, a *facto* of n and m is an array of integers such that the product of every element in the array is n, and the length of the array is m.

For example, when $n = 8, m = 2$, one of the *facto* is $[4, 2]$, as $4 \cdot 2 = 8$, and the array's length is 2. Other examples of *facto* for $n = 8, m = 2$ include $[-4, -2], [1, 8]$, and $[8, 1]$.

Codey wants you to find the total number of **unique** *facto* that exist for the n and m modulo $10^9 + 7$. Recall that a modulo b is the remainder of a when divided by b.

Input Format

The first line contains two integers, n and m, where n represents the number n, and m represents the number of *facto*.

Constraints

$1 \leq n, m \leq 10^6$

Output Format

Output the total number of unique *facto* for n and m modulo $10^9 + 7$.

Sample Test Case

`8 2`	`8`

facto for $n = 8, m = 2$ includes: $[1, 8]$, $[8, 1]$, $[-1, -8]$, $[-8, -1]$,$[4, 2]$, $[2, 4]$, $[-2, -4]$, $[-4, -2]$

Codey and Zoey

Problem Statement

Codey came to realize that there was something far more profound than simply solving problems, and that was the power of love. On this particular day, Codey decided to go on a date with Zoey.

The two of them found themselves at the origin $(0, 0)$ of a 2D grid, with their romantic destination set at (n, m) (cell in the n-th row and m-th column). Zoey, a fellow enthusiast for problem-solving, had no intention of making this date a walk in the park. She wanted to put Codey to the test to see if he was truly the one.

So, Zoey asked Codey to find the total number of ways for both of them to reach the destination (n, m) from the origin $(0, 0)$ while **intersecting** exactly k times modulo $10^9 + 7$, all while meeting the following conditions:

- They both take the shortest path.
- They move one step at a time.
- They cannot stop along the way.

Codey is eager to impress Zoey, but once again, it finds itself stuck on this problem. It requires your assistance in finding the number requested by Zoey. Help poor Codey out! Recall that a modulo b is the remainder of a when divided by b.

Input Format

The first line contains three positive integers n, m, k, which represent the destination of Codey and Zoey and the total number of intersections.

Constraints

- $0 \leq n, m \leq 50$, $(n, m) \neq (0, 0)$
- $2 \leq k \leq n + m + 1$

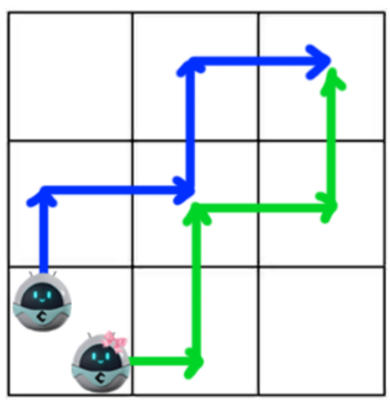
Output Format

Output the total number of ways for both of them to reach the destination (n, m) from the origin $(0, 0)$ while intersecting exactly k times modulo 10^9+7.

Sample Test Case 1

`2 2 3`	`12`

One of the ways for Codey and Zoey to intersect 3 times is shown below:



The 3 intersections in this example is $(0, 0), (1, 1), (2, 2)$.