Assignment-3

Computer Programming

Due Date: 8 Oct, 2023 11:59 PM

Important Notes

- OJ score will be considered as the final score.
- Plagiarism policy: Zero for the assignment if caught for plagiarising.

Question 1:

Once upon a time in the quaint town of Letterland, there was a special tradition. Every year, the townspeople celebrated the arrival of spring by crafting enchanting strings made up of English alphabets. These strings were called "**beautiful strings**".

To be considered a truly beautiful string, a few criteria had to be met:

1. It had to include at least one majestic **uppercase English letter**, representing the grandeur of the season.
2. It must also feature a humble **lowercase English letter**, signifying the gentleness of the blooming flowers.
3. Every character in the string had to be **unique** like the individual petals of a blossoming rose.

For instance, **`Charm`** and **`bEaUtiful`** qualify as beautiful strings, whereas **`Gorgeous`** and **`alluring`** do not

The townsfolk took great pride in creating these beautiful strings, for they believed it brought them good fortune and prosperity in the coming year.

Your task is to determine if a given string S possesses the magic to be labeled a beautiful string, and thus, continue this cherished tradition.

Input Constraints:

- $1 \leq |S| \leq 100$

Input:

The only line of input contains the string S

Output:

If S is a **beautiful string** print `Yes` otherwise print `No`

Sample Input 1:

```
Explain
```

Sample Output 1:

```
Yes
```

Sample Input 2:

```
Aa
```

Sample Output 2:

```
Yes
```

Sample Input 3:

```
Perfect
```

Sample Output 3:

```
No
```

Question 2:

While you were casually hanging out at the Juice Canteen, you bumped into your friend Kushagra. He posed an intriguing challenge. He handed you a string and asked if you could make as many different words as possible using its characters.

Excited to give it a try, you got to work. It felt a bit like solving a puzzle, and finding new ways to arrange the letters. Each arrangement revealed a new and interesting word. You eagerly dove into the task, ready to discover all the different words you could create with those given letters.

Input Constraints:

- $1 \leq n \leq 8$

Input:

The only input line has a string of length n. Each character is between a–z.

Output:

First print an integer k: the number of strings.

Then print k lines: the strings in alphabetical order.

Sample Input 1:

```
aabac
```

Sample Output 1:

```
20
aaabc
aaacb
aabac
aabca
aacab
aacba
abaac
abaca
abcaa
acaab
acaba
acbaa
baaac
baaca
bacaa
bcaaa
caaab
caaba
cabaa
cbaaa
```

Question 3:

A secret code enthusiast named Ayan has a liking for what he calls "*special*" strings. These strings are composed of lowercase English letters and have a length of n.

He defines the "*cipher distance*" between two letters as the numerical difference between their positions in a unique encryption scheme. For example, `cd(c,e) = cd(e,c) = 2` and `cd(a,z) = cd(z,a) = 25`.

Furthermore, the cipher distance between two special strings is calculated as the sum of the cipher distances between their respective letters. For instance, `cd(hat, daw) = cd(h,d) + cd(a,a) + cd(t,w) = 4 + 0 + 3 = 7`, and `cd(bear, roar) = 16 + 10 + 0 + 0 = 26`.

Ayan hands you a special string s and an integer k, throwing down the gauntlet to find any other special string s' such that `cd(s, s') = k`. However, if it's impossible to create such a string, please respond with `-1`.

Input Constraints:

- $1 \le n \le 10^5$
- $0 \le k \le 10^6$

Input:

The first line contains two integers n and k.

The second line contains a string s of length n, consisting of lowercase English letters.

Output:

If there is no string satisfying the given conditions then print `-1`.

Otherwise, print any nice string s' that cd(s, s') = k.

Sample Input 1:

```
4 26
bear
```

Sample Output 1:

```
roar
```

Sample Input 2:

```
2 7
af
```

Sample Output 2:

```
db
```

Sample Input 3:

```
3 1000
hey
```

Sample Output 3:

```
-1
```

Question 4

Plot: Batman has to leave Gotham urgently, and Andrea Botez the Gremlin is using this opportunity to try and take over the city. To the *knightmare* of the citizens their only protector now is Levy Rozman. The only way for levy to protect the city from Andrea is to reach Batman's safehouse, but Andrea's minions will try to intercept him before he can reach the safehouse. Although Levy has friends, no one wants to drink a potion with his dyed blonde hair. Can you help Levy reach the safehouse?

Statement: The city of Gotham can be seen as an n by m grid of buildings. Levy has Batman's "Knighthopper", a vehicle which can leap across the rooftops of buildings in the shape of a knight move.

All the buildings in the bottom row are entrances to the safehouse, and Levy can start anywhere in the first row and must reach an entrance in the bottom row. Levy can move across the grid in the shape of a knight move but because Andrea's minions are following him, he can only move down the grid. Andrea's minions are also stationed at k distinct squares $(x_1, y_1), (x_2, y_2), \ldots, (x_k, y_k)$ (not the starting squares or the safehouse entrances) on the grid, so Levy must avoid those squares.

Levy needs to choose which entrance he wants to reach before he starts moving, and he wants to choose the entrance which has the most paths leading to it. A path is a sequence of moves which starts at the top row and ends at that entrance.

Formally: Given an $n \times m$ grid, find an i, where $1 \le i \le m$ such that the number of valid paths starting from $(1, j)$ and ending at (n, i) summed over all j from 1 to m, is maximum, and print the total number of paths to (n, i) modulo $10^9 + 7$. If there are multiple such i, print the smallest one. A valid path is a sequence of knight moves with the condition that if one move is from (x, y) to (x', y') then $x' > x$, and $(x', y') \ne (x_r, y_r)$ for all $1 \le r \le k$. In this problem the possible knight moves from (x, y) are to $(x+1, y+2)$, $(x+1, y-2)$, $(x+2, y+1)$ and $(x+2, y-1)$.

Input Constraints:

- $2 \le n, m \le 1000$
- $1 \le k \le \min(n \cdot m, 10^5)$
- $2 \le x_i \le n-1,\ 1 \le y_i \le m$ for $1 \le i \le k$

Input:

The first line has three space-separated integers n, m, k

The next k lines have two space separated integers x_i and y_i

Output:

Print i which is the column of the entrance with the maximum number of total valid paths, followed by the total number of valid paths to i modulo $10^9 + 7$. If there are multiple possible values for i, print the smallest among them.

Sample Input 1:

```
3 4 1
2 4
```

Sample Output 1:

```
3 3
```

Question 5:

You need to know 3 more things about Kitansh Mayathwal to solve this problem. First is that Kitansh Mayathwal has N friends living in numbered houses on a straight line from 1 to N. Second is that each one of Kitansh Mayathwal's friends have one number X_i, which represents the amount they are irritated by him.

Third is that Kitansh Mayathwal can do anything.

What Kitansh Mayathwal does is -

- He chooses a House i $(1 \le i \le n)$ and goes to that friends house.
- While $i \le n$, the i_{th} friend then opens the door, sees Kitansh Mayathwal, and sends him X_i houses away to the right i.e. Kitansh Mayathwal will go to house $i + X_i$ (if the house exists). Kitansh Mayathwal considers himself to have done X_i amount of irritation from this step.
- This process of being sent away continues, and when $i > n$ Kitansh Mayathwal stops.

Kitansh Mayathwal wants to be as irritating as possible, so he wants to reach the maximum possible sum of irritations. Figure out which starting index Kitansh Mayathwal will start from to achieve this.

And as to why would he do this, refer to Point Number 3, on things needed to solve this problem.

Input:

The first line contains one integer t $(1 \le t \le 10^4)$ — the number of test cases. Then t test cases follow.

The first line of each test case contains one integer n $(1 \le n \le 2 \times 10^5)$ — the number of friends Kitansh Mayathwal has.

The next line contains n integers $x_1, x_2, \ldots, x_n$ $(1 \le x_i \le 1e9)$ - the amount each friend is irritated by Kitansh Mayathwal.

It is guaranteed that the sum of n over all test cases does not exceed 2×10^5.

Output:

For each test case, output on a separate line one number — the maximum irritation that Kitansh Mayathwal can achieve. Kitansh chooses the starting position from 1 to n to maximise his irritation.

Sample Input 1:

```
3
6
8 1 3 2 2 3
4
2 1 4 1
3
1 1 1
```

Sample Output 1:

```
8
6
3
```