CS F211
Data Structures and Algorithms
Assignment - 4

Allowed languages: **C**

February 11, 2021

General Tips

- Try to use functions as much as possible in your code. Functions increase reusability and the pass-by-value feature provides a significant help sometimes. Modularizing your code also helps you to debug efficiently.
- Use `scanf` to read characters/strings from STDIN. Avoid using `getchar`, `getc` or `gets`. Try to read up about character suppression in `scanf` as it will be very helpful in some of the problems.
- Use `printf` instead of `putc`, `putchar` or `puts` to print character/string output on STDOUT.
- Indent your code appropriately and use proper variable names. These increase readability and writability of the code. Also, use comments wherever necessary.
- Use a proper IDEs like Sublime Text or VSCode as they help to run and test your code on multiple test-cases easily. You can install Windows Subsystem Linux (WSL) or MinGW 7.3.0, if you are Windows user to compile and run your programs. Alternatively, you can run and test your codes on Online GDB. If you are using WSL or Linux to run your programs, make sure that the gcc version is `gcc 5.4.1 c99`.

A: Sky of a Million Stars

A cave discovered on Mars appears to have a bunch of inscriptions in the wall, presumably made by an alien race. Linguists and anthropologists who are studying the images beamed back to Earth realised that they could piece together some of the information if they knew the order of that foreign alphabet. The alien race, surprisingly, uses a **subset** of the English alphabet [A-Z], but with a different alphabets order. Experts studying the inscriptions could discover N rules about the alien alphabetical order. Each rule is represented as c_1 c_2, meaning letter c_1 comes before c_2 in the alien alphabet. Note that all alphabets that are a part of the alien alphabet **will definitely** appear atleast once in the rules. Print **one** valid order of the alien alphabet. If there are multiple alien alphabet orders possible, print any one. If the alien language is self-contradictory, print "ALIENS BE CRAZY" and exit.

Input

The first line contains an integer N ($1 \leq N \leq \binom{26}{2}$). The next N lines contain two characters each (space-separated), representing one rule. It is guaranteed that all characters in rules are capital letters A-Z.

Output

The order of the alien alphabet in one line, with no spaces. If no order is possible, print the string "ALIENS BE CRAZY".

---

input
```
3
X B
C X
C A
```

output
```
CAXB
```

explanation
Other possible answers are CXAB, CXBA

input
```
8
R E
A S
B A
E A
L M
A R
B X
X A
```

output
```
ALIENS BE CRAZY
```

explanation
No valid outputs possible.

B: The Three Laws

A robot may not injure a human being, or, through inaction, allow a human being to come to harm.

In a bid to garner publicity for their new line of security and police robots, U.S. Robots is organising a RoboWars event that will be televised on international TV. The tournament has N robots participating, each with strength R_i. When a robot with strength R_i fights with a robot of strength R_j, the robot with higher strength wins, and destroys the weaker robot. The weaker robot is completely destroyed and eliminated from the tournament. The stronger robot (say, R_i) survives, but it's strength changes to $R_i(new) = abs(R_i - \alpha(R_i - R_j))$, where $0 \leq \alpha \leq 10$. If two robots are equally matched, they both die in each others hand. The tournament consists of a number of rounds, which continue till only one (or none) robots are left standing. The robots are initially placed in a straight line, and in the first round, the first robot fights the second, the third robot fights the fourth and so on. If a robot is leftover at the end of the line, and can't be paired it doesn't take part in the round, and recharges while everyone else fights, allowing it to increase its strength by β in that round. Once the round is complete, dead robots are removed and the next round commences. Determine the position and final strength of the winner.

Input

The first line contains three space separated integers - N ($0 \leq N \leq 10^5$), α ($0 \leq \alpha \leq 10$) and β ($500 \leq \beta \leq 10^3$). The next line contains N space separated integers numbers, denoting the strengths of each of the robots in line. ($0 \leq R_i \leq 10^4 \; \forall i$).

Output

Print two space separated numbers i and R_f. i is the position the final surviving robot (winner of the tournament) was standing in at the start of the tournament (the initial line is 1-indexed). R_f is the strength of the winner after the final round. If no robots are left standing print $-1 - 1$.

---

input
5 5 30
70 75 80 80 20
output
-1 -1
explanation
After the first round, the robot strengths are: {50, 50} - which leads to both of them dying in the final round. No robot left standing..

input
7 8 40
70 60 50 120 100 50 40
output
4 18280
explanation
After round 1: {10, 440, 300, 80}
After round 2: {3000, 340}
After round 3: {18280}

C: Talking to Myself

Mike likes playing a word game with himself, where he tries to transform a word, S (length M), to a different word E (length M). He begins by changing (replacing) one letter of the S, to form an intermediate word W_1. He then changes another letter in W_1 to form W_2. He does this to form a sequence of words $S, W_1, W_2 \ldots W_L, E$. The end word is special, since it must satisfy the property $E[i] \neq S[i]$, $\forall i$ such that $0 \leq i \leq M-1$. Additionally, he cannot arbitrarily replace any letter with another letter - all intermediate words (W_i) and the end word (E) must be real words, which means they must be in a dictionary D (containing N words). Determine if such an E exists, and if it does find the *minimum* number of intermediate words required to get to an end word E from the start word S. If it is impossible to do so, print -1. If there are multiple possible words E, choose the one that has a minimum number of intermediate steps. *Additional thinking: if you had to implement this for the entire English dictionary, would your approach work?*

Input

The first line of the input contains integer N such that $1 \leq N \leq 1000$. The second line of input contains the word S. The next N lines contain one word each (D_i) such that $1 \leq len(D_i) \leq 12$ representing all words in the dictionary.

Output

Output a single integer, denoting the **minimum** number of intermediate words you'd need to generate before reaching an end word.

---

input 1

```
9
app
xi
ben
ape
apo
xu
ave
bpo
beo
ove
```

output 1

```
2
```

input 2

```
9
rome
tome
some
hell
hall
peep
xhud
yyee
yuii
iyll
```

output 2

```
-1
```

explanation 1

app -> ape -> ave -> ove. ove is an end word, by definition, and we need two intermediate words to get here. app -> apo -> bpo -> beo -> ben is also a valid path, with end word "ben", but this has 3 words.

D: Wikipedia

TIME Person of the Year, 2006

Rohan began reading a Wikipedia article S initially. In each Wikipedia article, there are links to other Wikipedia articles. Being the curious lad he is, one of the links **might** interest him, causing him to click on it and read the linked article. However, he doesn't pick what article to choose out of the links at random - for each link in an article, there is a certain probability (p_{ij}) that Rohan will click it. There is also a probability (q_i) that he will not click on any link at all, and just close the browser after reading the current article. This probability q_j for a certain article, can be calculated as $q_j = 1 - \sum_i p_{ij}$, where p_{ij} is summed over all outgoing links on that article. Some articles might not have any outgoing links, meaning he will shut down his browser after reading that last article. Given a set of links (E) and probabilities associated with clicking each link, calculate, for each article, the probability that it is the last article he reads. It is guaranteed that articles (W_i) and edges (E) form a tree rooted at S. Rank all articles from highest to lowest probabilities of it being the last article read.

Input

The first line has integers N, E, and S. The next E lines contain three space separated numbers - (W_i, W_j, p_{ij}) - meaning that article W_i has a link to W_j, and there is a p_{ij} probability that this link will be clicked.

Output

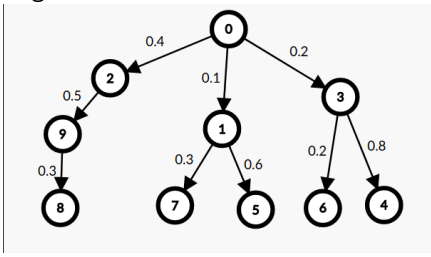
Print N space separated numbers, the articles ranked in descending order of probabilities. If two articles have the same probability, order them in increasing order of article number.

---

input

```
10 9 0
0 1 0.1
0 2 0.4
0 3 0.2
1 7 0.3
1 5 0.6
2 9 0.5
9 8 0.3
3 4 0.8
3 6 0.2
```

figure



output

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

explanation

Probabilities for ending on articles $\{0..9\}$ are {0.3, 0.01, 0.2, 0, 0.16, 0.06, 0.04, 0.03, 0.06, 0.14} respectively. Article 0, which has the highest probability (0.3), is printed first. Articles 2, 4, 9 have the next three highest possibilities (0.2, 0.16, and 0.14). The next highest probability of 0.06 is shared by articles 5 and 8 - 5 must be printed before 8, as $5 < 8$.

E: Connecting Nails

On a wooden plank, you have M nails driven into the wood at points $S = \{(x_1, y_1) \dots (x_M, y_M))\}$, with each nail being one point. Nails are connected in this order: sweep a reference line from the X axis upwards (order of quadrants: Q1, Q2, Q3, Q4), completing a full turn while joining every subsequent point. Let S_0, S_1 be the first two vertices encountered while sweeping the line, and let d_0 be the Manhattan Distance between them. Calculate the sum of all pairwise Manhattan distances ($\sum_i d_i$) between all such pairs of points encountered as the reference line sweeps all four quadrants It is guaranteed that, for any two distinct points $I(x_i, y_i)$, $J(x_j, y_j)$, I, J and the origin $O(0, 0)$ are never collinear if I and J are in the same quadrant.

Input

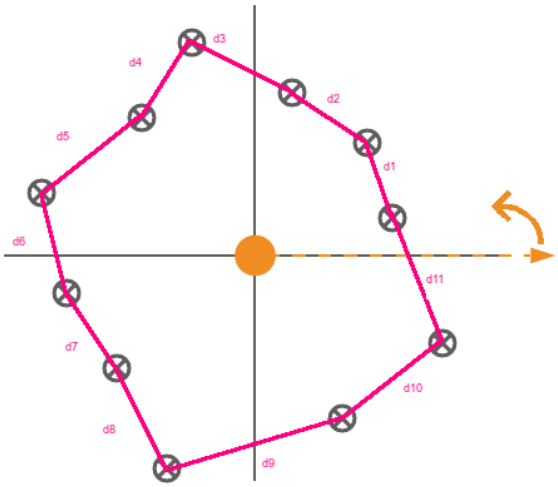
The first line contains the number M. The next M ($3 \leq M \leq 10^5$) lines contain two space separated integers representing x_i, y_i ($-1.5 * 10^5 \leq x_i, y_i \leq 1.5 * 10^5$).

Output

Print one integer, L, the sum of all pairwise Manhattan distances.

---

figure



The answer to the above problem is the sum of all Manhattan distances of the points connected by red line segments, i.e. $\sum_i d_i$. Orange arrow indicates the beginning point, and the direction to rotate our reference line.

input
```
3
4120 0
-1000 9800
-3210 -6467
```

output
```
47194
```

input
```
4
-1 6
1 1
4 -3
-1 -2
```

output
```
28
```

explanation
Order of points: $(1, 1)$, $(-1, 6)$, $(-1, -2)$, $(4, -3)$, $(1, 1)$

F: Polifia

Tim is a guitar player and wants to hold a concert. The venue can be represented by a cartesian co-ordinate plane. Tim has N amplifiers where the i^{th} amplifier has a power output of a_i watts and is located at the position (x_i, y_i). Each of these N amplifiers has already been placed in the venue by the sound engineer and cannot be moved. There is something very peculiar about this particular venue: The people in the crowd can only stand along *some line parallel to* $y = x$. Note that this line is chosen by Tim and can be placed anywhere in the co-ordinate plane. Tim wants the audience to get an equal volume of sound from both sides so he wants to place the line such that the *sum of powers of amplifiers on both sides of the line is equal*. Help Tim find if placing such a line is possible. More formally, you must find if it is possible to choose a line $y = x + c$ where c is a variable such that sum of powers of amplifiers on both sides of the line is equal.

Input

The first line of input contains one integer T, the number of testcases. For each testcase, there will be $N + 2$ lines of input as described below. The first line of each testcase consists of N $(1 \leq N \leq 10^5)$, the number of amplifiers. The second line contains N integers representing the powers of the amplifiers. The next N lines contain two integers each: x_i and y_i, the co-ordinates of the i^{th} amplifier $(-10^9 \leq x_i, y_i \leq 10^9)$

Output

Print T lines, with each line containing either YES or NO in all capitals. (please print in all capitals to avoid problems with Mooshak)

---

input

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

output

```
YES
```

explanation

We have one testcase. Any line of the form $y = x + c \ \forall -2 < c < 2$ would divide the amplifiers such that both sides have a total power of 4 watts

---

G: Tuul

Danny Carey has N drum pads lined up in a row, where the i^{th} drum pad has a frequency a_i. For the next song Danny needs all the drums pads to be arranged in a non-decreasing order of frequencies. In one swap, Danny can swap the positions of two adjacent drum pads. He needs to quickly tell his drum mechanic the minimum number of such swaps required to arrange the drum pads in a non-decreasing order of frequencies. Despite being a natural at polyrhythms, he was too slow at this task so he asked you for help.
Hint: Try and think of a divide and conquer strategy

Input

The first line of input contains a single integer N ($1 \leq N \leq 10^5$), the number of drum pads. The second line contains N integers, where a_i represents the frequency of the i^{th} drum pad ($-10^9 \leq a_i \leq 10^9$).

Output

output a single integer, the minimum number of swaps required to arrange the drums as described above.

```
input
5
1 20 6 4 5

output
5

explanation
first swap 20 with 6, then 20 with 4, then 20 with 5.  Then swap 6 with 4 and then
6 with 5.  With a total of 5 swaps, we get the arrangement 1 4 5 6 20
```

H: Meshuguhh

Fredrik Thordendal wants to make an ironic song using only the major scale to confuse his fans. The major scale consists of 7 notes, represented by the upper case letters A to G (inclusive). He writes out a riff for this (in drop F tuning obviously) using these notes. Here, a riff is a sequence of notes where each note is a capital letter from the set of letters A to G (inclusive). The drummer Thomas noticed that the riff was way too long so he asked Frederik to choose the *smallest possible contiguous subsegment* from the riff that contains *all* the types of notes present in the riff, and use that for the song. Can you help Frederik find the length of such a subsegment? More formally, your task is to find the smallest subsegment such that the set of distinct notes in the subsegment should be equal to the set of distinct notes in the entire riff.

Input

The first line contains a single integer N, the number of notes in the entire riff ($1 \leq N \leq 10^5$). The second line contains a string of characters representing the riff itself. It is guaranteed that the string only consists of upper case english letters from A to G (inclusive).

Output

print a single integer, the length of the smallest possible contiguous subsegment that contains all the types of notes contained in the original riff

---

input
9
ABBGDCACD

output
5

explanation
We can take the subsegment from index 3 to index 7 (inclusive). This subsegment includes the notes B,G,D,C,A which contains all the notes present in the entire riff

---

I: Melattica

Lars Ulrich isn't a very good drummer so he decided to use a Machine Learning model to write drum parts for him. But now the rest of the band likes the machine learning model better and decides to fire Lars. Now he has to use his machine learning skills in other ways. He develops a model that can check if a permutation is in increasing order or not but he's not sure if the model actually works. To check this, he decided to perform a test: he takes a number x and expects it to be at position *pos* if the permutation is sorted. He then performs a binary search on the permutation and if the number x is at position *pos* according to the binary search, then he says that the permutation is increasing. (Recall that a permutation of size N is an array such that every number from 1 to N occurs exactly once). The C code for the binary search he performs is given below

```
int l=0, r=n-1;
while(l<r) {
    mid = (l+r)/2;
    if(arr[mid]<=x) l = mid+1;
    else r = mid;
}
l>0 && a[l-1]==x? printf("Yes") : printf("No");
```

But what Lars doesn't know is that the permutation need not be necessarily sorted to find the number x at position *pos*. Can you find how many such permutations exist such that the binary search algorithm finds x at the position *pos*? Since the number can be large, print the answer modulo $10^9 + 7$
source for those interested: NSynth, using neural networks to create melodies

Input

The first line contains three numbers, N, x, and *pos* ($1 \leq x \leq N \leq 1000$, $0 \leq pos \leq N - 1$).

Output

Print a single integer, the number of possible permutations modulo $10^9 + 7$

---

input
4 1 2

output
6

explanation
the possible permutations are (2,3,1,4) , (2,4,1,3), (3,2,1,4), (3,4,1,2), (4,2,1,3), (4,3,1,2)

J. Zero Selects

You are giving the second round of inductions for the music club and the keyboardist plays a "arpeggio" of N notes consecutively on his keyboard. You can tell from the sound that all the notes first increase to a certain point and then decrease till the end (Note that there can be 0 increasing or 0 decreasing notes). You are able to figure out the sequence of notes he plays and have stored that sequence in your memory where the i^{th} note played is a_i. The keyboardist then asks you q questions, where in the i^{th} question he asks if he played the note t_i **anywhere** in the sequence. You have to correctly answer YES or NO for all queries or you'll have to wait another semester to get into the club! To help you out with this refer to this link on ternary search

Input

The first line contains 2 integers N and q ($1 \leq N, q \leq 10^5$), the number of notes played and the number of questions asked. The second line contains N integers, representing the notes played. It is guaranteed that the notes first increase and then decrease as described above. The next q lines contain one integer each: t_i where t_i represents the i^{th} question ($0 \leq t_i \leq 10^9$).

Output

output q lines, each containing a single word YES or NO in all capitals. (Please use all capitals to avoid problems with mooshak)

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

output
YES
YES
NO
NO
```
