A - Chocolate

Time Limit: 2 sec / Memory Limit: 1024 MiB

Score: 500 points

Problem Statement

Ms. AtCoder has decided to distribute chocolates to N friends on Valentine's Day. For the i-th friend $(1 \leq i \leq N)$, she wants to give a square chocolate bar of size $2^{A_i} \times 2^{A_i}$.

She has procured a rectangular chocolate bar of size $H \times W$. It is partitioned by lines into a grid of H rows and W columns, each cell being a 1×1 square.

Determine whether it is possible to divide the chocolate bar along the lines into several pieces to obtain all the chocolate bars for her friends. It is fine to have leftover pieces.

Constraints

- $1 \leq H \leq 10^9$
- $1 \leq W \leq 10^9$
- $1 \leq N \leq 1000$
- $0 \leq A_i \leq 25$ $(1 \leq i \leq N)$
- All input values are integers.

Input

The input is given from Standard Input in the following format:

$$
\begin{array}{l}
H \quad W \quad N \\
A_1 \quad A_2 \quad \cdots \quad A_N
\end{array}
$$

Output

If the objective is achievable, print `Yes`; otherwise, print `No`.

Sample Input 1

```
4 4 4
1 0 0 1
```

Sample Output 1

```
Yes
```

By dividing a 4×4 chocolate bar as shown in the figure below, you can obtain pieces of size $2 \times 2, 1 \times 1, 1 \times 1, 2 \times 2$.

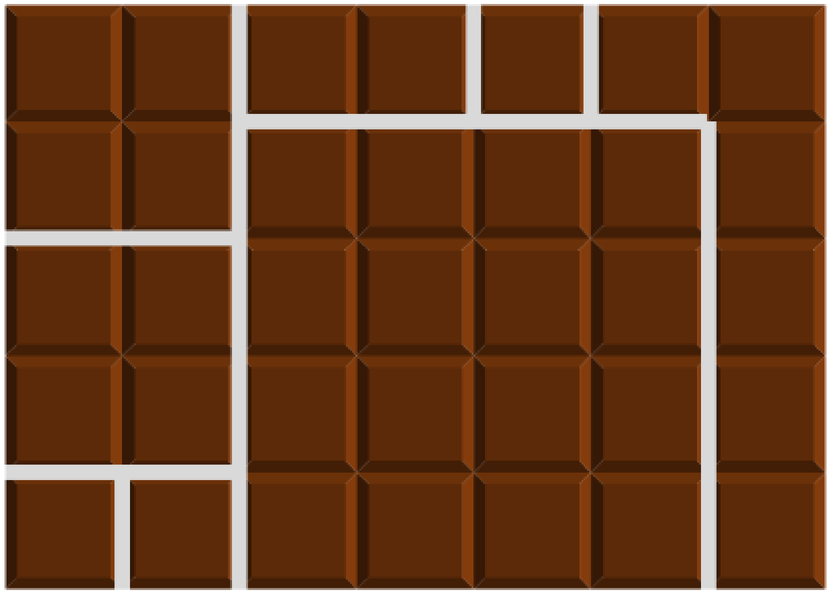
Sample Input 2

```
5 7 6
0 1 0 2 0 1
```

Sample Output 2

```
Yes
```

By dividing a 5×7 chocolate bar as shown in the figure below, you can obtain pieces of size $1 \times 1, 2 \times 2, 1 \times 1, 4 \times 4, 1 \times 1, 2 \times 2$.



Sample Input 3

```
3 2 7
0 0 0 0 0 0 0
```

Sample Output 3

```
No
```

It is impossible to obtain seven pieces of size 1×1 from a 3×2 chocolate bar.

Sample Input 4

```
11 11 2
2 3
```

Sample Output 4

```
No
```

It is impossible to obtain both a 4×4 and an 8×8 piece from an 11×11 chocolate bar.

Sample Input 5

```
777 777 6
8 6 9 1 2 0
```

Sample Output 5

```
Yes
```

B - AtCoder Language

Time Limit: 2 sec / Memory Limit: 1024 MiB

Score: 500 points

Problem Statement

The AtCoder language has L different characters. How many N-character strings s consisting of characters in the AtCoder language satisfy the following condition? Find the count modulo 998244353.

- All K-character subsequences of the string s are different. More precisely, there are ${}_N\mathrm{C}_K$ ways to obtain a K-character string by extracting K characters from the string s and concatenating them without changing the order, and all of these ways must generate different strings.

▶ What is ${}_N\mathrm{C}_K$?

Constraints

- $1 \leq K < N \leq 500000$
- $1 \leq L \leq 10^9$
- All input values are integers.

Input

The input is given from Standard Input in the following format:

$$N \quad K \quad L$$

Output

Print the count modulo 998244353.

Sample Input 1

```
4 3 2
```

Sample Output 1

```
2
```

If `a` and `b` represent the first and second characters in the language, the condition is satisfied by two strings: `abab` and `baba`.

Sample Input 2

```
100 80 26
```

Sample Output 2

```
496798269
```

Approximately 10^{86} strings satisfy the condition, but here we print the count modulo 998244353, which is 496798269.

Sample Input 3

```
100 1 26
```

Sample Output 3

```
0
```

Sample Input 4

```
500000 172172 503746693
```

Sample Output 4

```
869120
```

C - Election

Time Limit: 2 sec / Memory Limit: 1024 MiB

Score: 600 points

Problem Statement

This year's AtCoder mayoral election has two candidates, A and B, and N voters have cast their votes. The voters are assigned numbers from 1 to N, and voter i $(1 \leq i \leq N)$ voted for candidate c_i.

Now, the votes will be counted. As each vote is counted, the provisional result will be announced as one of the following three outcomes:

- **Result A:** Currently, candidate A has more votes.
- **Result B:** Currently, candidate B has more votes.
- **Result C:** Currently, candidates A and B have the same number of votes.

There is a rule regarding the order of vote counting: votes from all voters except voter 1 must be counted in ascending order of their voter numbers. (The vote from voter 1 may be counted at any time.)

How many different sequences of provisional results could be announced?

▶ What is a sequence of provisional results?

Constraints

- N is an integer such that $2 \leq N \leq 1000000$.
- Each of $c_1, c_2, \ldots, c_N$ is `A` or `B`.

Input

The input is given from Standard Input in the following format:

$$N$$
$$c_1 c_2 \cdots c_N$$

Output

Print the answer.

Sample Input 1

```
4
AABB
```

Sample Output 1

```
3
```

In this sample input, there are four possible orders in which the votes may be counted:

- The votes are counted in the order of voter $1 \to 2 \to 3 \to 4$.
- The votes are counted in the order of voter $2 \to 1 \to 3 \to 4$.
- The votes are counted in the order of voter $2 \to 3 \to 1 \to 4$.
- The votes are counted in the order of voter $2 \to 3 \to 4 \to 1$.

The sequences of preliminary results for these will be `AAAC`, `AAAC`, `ACAC`, `ACBC` from top to bottom, so there are three possible sequences of preliminary results.

Sample Input 2

```
4
AAAA
```

Sample Output 2

```
1
```

No matter the order of counting, the sequence of preliminary results will be `AAAA`.

Sample Input 3

```
10
BBBAAABBAA
```

Sample Output 3

```
5
```

Sample Input 4

```
172
AABAAAAAABBABAABBBBAABBAAABBABBABABABBAAABAAABAABAABBBBABBBABBABBBBBBBBAAABAAABAAABABBBAABAAAAABABBABBA
BBBBBABAABAABBBABABBAAAABAABABBBABAAAABBBBABBBABBBABAABBBAAAABAAABAAAB
```

Sample Output 4

```
24
```

D - Distance Ranking

Time Limit: 2 sec / Memory Limit: 1024 MiB

Score: 700 points

Problem Statement

Place N points $p_1, p_2, \dots, p_N$ in an N-dimensional space to satisfy the following conditions:

> **Condition 1** The coordinates of the points consist of integers between 0 and 10^8, inclusive.
>
> **Condition 2** For $(A_1, B_1), (A_2, B_2), \dots, (A_{N(N-1)/2}, B_{N(N-1)/2})$ specified as input, $d(p_{A_1}, p_{B_1}) < d(p_{A_2}, p_{B_2}) < \dots < d(p_{A_{N(N-1)/2}}, p_{B_{N(N-1)/2}})$ must hold. Here, $d(x, y)$ denotes the Euclidean distance between points x and y.

It can be proved that a solution exists under the constraints of the problem. If multiple solutions exist, just print one of them.

▶ What is Euclidean distance?

Constraints

- $3 \le N \le 20$
- $1 \le A_i < B_i \le N \ (1 \le i \le \frac{N(N-1)}{2})$
- All pairs $(A_1, B_1), (A_2, B_2), \dots, (A_{N(N-1)/2}, B_{N(N-1)/2})$ are distinct.

Input

The input is given from Standard Input in the following format:

$$
\begin{array}{l}
N \\
A_1 \quad B_1 \\
A_2 \quad B_2 \\
\vdots \\
A_{N(N-1)/2} \quad B_{N(N-1)/2}
\end{array}
$$

Output

Let $(p_{i,1}, p_{i,2}, \dots, p_{i,N})$ be the coordinates of point p_i. Print your solution in the following format:

$$
\begin{array}{llll}
p_{1,1} & p_{1,2} & \cdots & p_{1,N} \\
p_{2,1} & p_{2,2} & \cdots & p_{2,N} \\
\vdots & & & \\
p_{N,1} & p_{N,2} & \cdots & p_{N,N}
\end{array}
$$

Sample Input 1

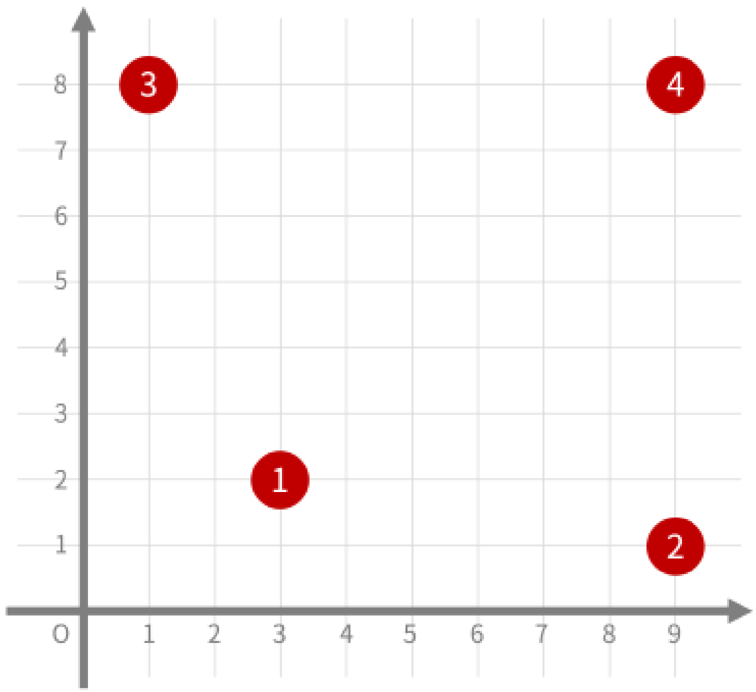
```
4
1 2
1 3
2 4
3 4
1 4
2 3
```

Sample Output 1

```
3 2 0 0
9 1 0 0
1 8 0 0
9 8 0 0
```

In this sample output, the third and fourth components of the coordinates are all zero, so the solution can be shown in the two-dimensional figure below.

$d(p_1, p_2) = \sqrt{37}, d(p_1, p_3) = \sqrt{40}, d(p_2, p_4) = \sqrt{49}, d(p_3, p_4) = \sqrt{64}, d(p_1, p_4) = \sqrt{72}, d(p_2, p_3) = \sqrt{113}$, and they are in the correct order.



E - Last 9 Digits

Time Limit: 2 sec / Memory Limit: 1024 MiB

Score: 800 points

Problem Statement

Determine whether there is a positive integer n such that the remainder of n^n divided by 10^9 is X. If it exists, find the smallest such n.

You will be given Q test cases to solve.

Constraints

- $1 \leq Q \leq 10000$
- $1 \leq X \leq 10^9 - 1$
- X is neither a multiple of 2 nor a multiple of 5.
- All input values are integers.

Input

The input is given from Standard Input in the following format, where X_i is the value of X in the i-th test case $(1 \leq i \leq Q)$:

$$
\begin{array}{l}
Q \\
X_1 \\
X_2 \\
\vdots \\
X_Q
\end{array}
$$

Output

Print Q lines. The i-th line should contain the answer for the i-th test case. If no n satisfies the condition, the answer should be -1.

Sample Input 1

```
2
27
311670611
```

Sample Output 1

```
3
11
```

This sample input consists of two test cases.

- The first case: The remainder of $3^3 = 27$ divided by 10^9 is 27, so $n = 3$ satisfies the condition.
- The second case: The remainder of $11^{11} = 285311670611$ divided by 10^9 is 311670611, so $n = 11$ satisfies the condition.

F - Walking

Time Limit: 2 sec / Memory Limit: 1024 MiB

Score: 1000 points

Problem Statement

The Kingdom of AtCoder has $2N$ intersections labeled with numbers from 1 to $2N$. Additionally, there are three types of **one-way** roads in the kingdom:

- **Type A:** For each $2 \leq i \leq N$, there is a road from intersection $2i-3$ to intersection $2i-1$.
- **Type B:** For each $2 \leq i \leq N$, there is a road from intersection $2i-2$ to intersection $2i$.
- **Type C:** For each $1 \leq i \leq N$, there is a road connecting intersection $2i-1$ and intersection $2i$ with direction s_i.

Here, s_i is X or Y, where direction X means the road goes from intersection $2i-1$ to intersection $2i$, and direction Y means it goes from intersection $2i$ to intersection $2i-1$.

Takahashi wants to **walk** several times so that for each $1 \leq i \leq N$, the direction of Type-C road connecting intersections $2i-1$ and $2i$ will be t_i.

What is a walk?

Start at any intersection and repeat the following action zero or more times:

- If it is possible to proceed to a Type-C road from the current intersection, walk along the Type-C road to the next intersection.
- Otherwise, if it is possible to proceed to a Type-A or B road from the current intersection, walk along that road to the next intersection.

Then, reverse the directions of all Type-C roads that were passed.

Calculate the minimum number of walks needed to achieve the goal and provide one method to achieve the goal in the minimum number of walks. Under the constraints of this problem, it can be proved that the goal can be achieved in a finite number of walks.

Constraints

- N is an integer such that $1 \leq N \leq 4000$.
- Each of $s_1, s_2, \dots, s_N$ is X or Y.
- Each of $t_1, t_2, \dots, t_N$ is X or Y.

Input

The input is given from Standard Input in the following format:

$$N$$
$$s_1 s_2 \cdots s_N$$
$$t_1 t_2 \cdots t_N$$

Output

Print your solution in the following format, where X is the number of walks, and the i-th walk $(1 \leq i \leq X)$ starts at intersection a_i and finishes at intersection b_i.

$$X$$
$$a_1 \quad b_1$$
$$a_2 \quad b_2$$
$$\vdots$$
$$a_X \quad b_X$$

Sample Input 1

```
5
XYXYX
XXYXX
```

Sample Output 1

```
1
2 7
```

In this sample input, Takahashi will walk as follows:

- Start at intersection 2.
- Since there is no Type-C road to proceed from intersection 2, walk along the Type-B road to intersection 4.
- Since there is a Type-C road to proceed from intersection 4, walk along the Type-C road to intersection 3.
- Since there is no Type-C road to proceed from intersection 3, walk along the Type-A road to intersection 5.
- Since there is a Type-C road to proceed from intersection 5, walk along the Type-C road to intersection 6.
- Since there is no Type-C road to proceed from intersection 6, walk along the Type-B road to intersection 8.
- Since there is a Type-C road to proceed from intersection 8, walk along the Type-C road to intersection 7.
- Finish at intersection 7.

This walk passes through the following three Type-C roads:

- The road connecting intersection 3 and intersection 4.
- The road connecting intersection 5 and intersection 6.
- The road connecting intersection 7 and intersection 8.

The directions of these roads are reversed, so the directions of the roads connecting intersections $2i - 1$ and $2i$ for $i = 1, 2, 3, 4, 5$ are now `X`, `X`, `Y`, `X`, `X`, respectively, achieving the goal.

Sample Input 2

```
5
XXYYX
XXYYX
```

Sample Output 2

```
0
```

Note that the goal is already achieved at the beginning, so there is no need to walk even once.

Sample Input 3

```
5
XXXXX
YYYYY
```

Sample Output 3

```
5
1 2
3 4
5 6
7 8
9 10
```

Sample Input 4

```
20
XXXYXYYXXXYXXXXYYXXY
XXYXYYXXYXXYXYXYXYXY
```

Sample Output 4

```
5
14 18
29 38
14 26
5 10
27 35
```

Sample Input 5

```
20
YXYXYXYYYXYYXYYXYXXX
XXXXXYXYYYXYYXXYYYXY
```

Sample Output 5

```
5
29 36
10 38
2 3
4 7
28 40
```