A - ABC Symmetry

Time Limit: 2 sec / Memory Limit: 1024 MiB

Score : 500 points

Problem Statement

For a non-empty string T consisting of A, B, and C, we call it a good string if it can be turned into an empty string by performing the following two types of operations any number of times in any order.

- Operation 1: Choose two identical characters in the string and delete them (cannot be performed if there are not two or more identical characters).
- Operation 2: Choose one A, one B, and one C in the string and delete them (cannot be performed if there are not one or more of each of A, B, and C).

For example, ABACA is a good string because it can be turned into an empty string by performing the operations as follows:

- Choose the 2nd, 4th, and 5th characters and delete them (Operation 2). The string becomes AA.
- Choose the 1st and 2nd characters and delete them (Operation 1). The string becomes an empty string.

You are given a string S of length N consisting of A, B, C, and ?. How many ways are there to replace each ? with A, B, or C to form a string that contains **at least** K good strings as contiguous substrings? Substrings are counted separately if they are at different positions in the original string, even if they are identical strings.

Find the count modulo 998244353.

Constraints

- $1 \leq N \leq 50$
- $0 \leq K \leq \frac{N(N+1)}{2}$
- N and K are integers.
- $|S| = N$
- S is a string consisting of A, B, C, and ?.

Input

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

```
N K
S
```

Output

Print the answer modulo 998244353.

Sample Input 1

```
4 2
A?AB
```

Sample Output 1

```
1
```

By replacing `?` with `A`, `B`, or `C`, we can obtain the following three strings: `AAAB`, `ABAB`, `ACAB`.

Among these, `AAAB` contains two good substrings: the `AA` at positions $1, 2$ and the `AA` at positions $2, 3$. Note that even if the substrings are identical as strings, they are counted separately if they are at different positions in the original string.

On the other hand, `ABAB` contains only one good substring `ABAB`. Also, `ACAB` contains only one good substring `CAB`.

Sample Input 2

```
50 411
??AB??C???????????????????????????????A???C????A??
```

Sample Output 2

```
457279314
```

Print the count modulo 998244353.

Sample Input 3

```
1 0
A
```

Sample Output 3

```
1
```

B - Symmetric Painting

Time Limit: 2 sec / Memory Limit: 1024 MiB

Score : 600 points

Problem Statement

On a circle, there are N equally spaced points numbered $0, 1, \ldots, N-1$ in this order, with Alice at point 0 and Bob at point K. Initially, all points are colored white. Starting with Alice, they alternately perform the following operation:

- Choose one of the currently white points and color it black. Here, after the operation, the coloring of the points must be symmetric with respect to the straight line connecting the operator and the center of the circle.

If the operator cannot perform an operation satisfying the above condition, the sequence of operations ends there.

Both players cooperate and make the best choices to maximize the total number of points colored black in the end. Determine whether all points are colored black at the end of the sequence of operations.

You are given T test cases to solve.

Constraints

- $1 \leq T \leq 10^5$
- $2 \leq N \leq 2 \times 10^5$
- $1 \leq K \leq N-1$
- All input values are integers.

Input

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

$$T$$
$$\mathrm{case}_1$$
$$\mathrm{case}_2$$
$$\vdots$$
$$\mathrm{case}_T$$

Each test case case_i $(1 \leq i \leq T)$ is in the following format:

$$N \quad K$$

Output

Print T lines. The i-th line should contain `Yes` if all points can be colored black for the i-th test case, and `No` otherwise.

Sample Input 1

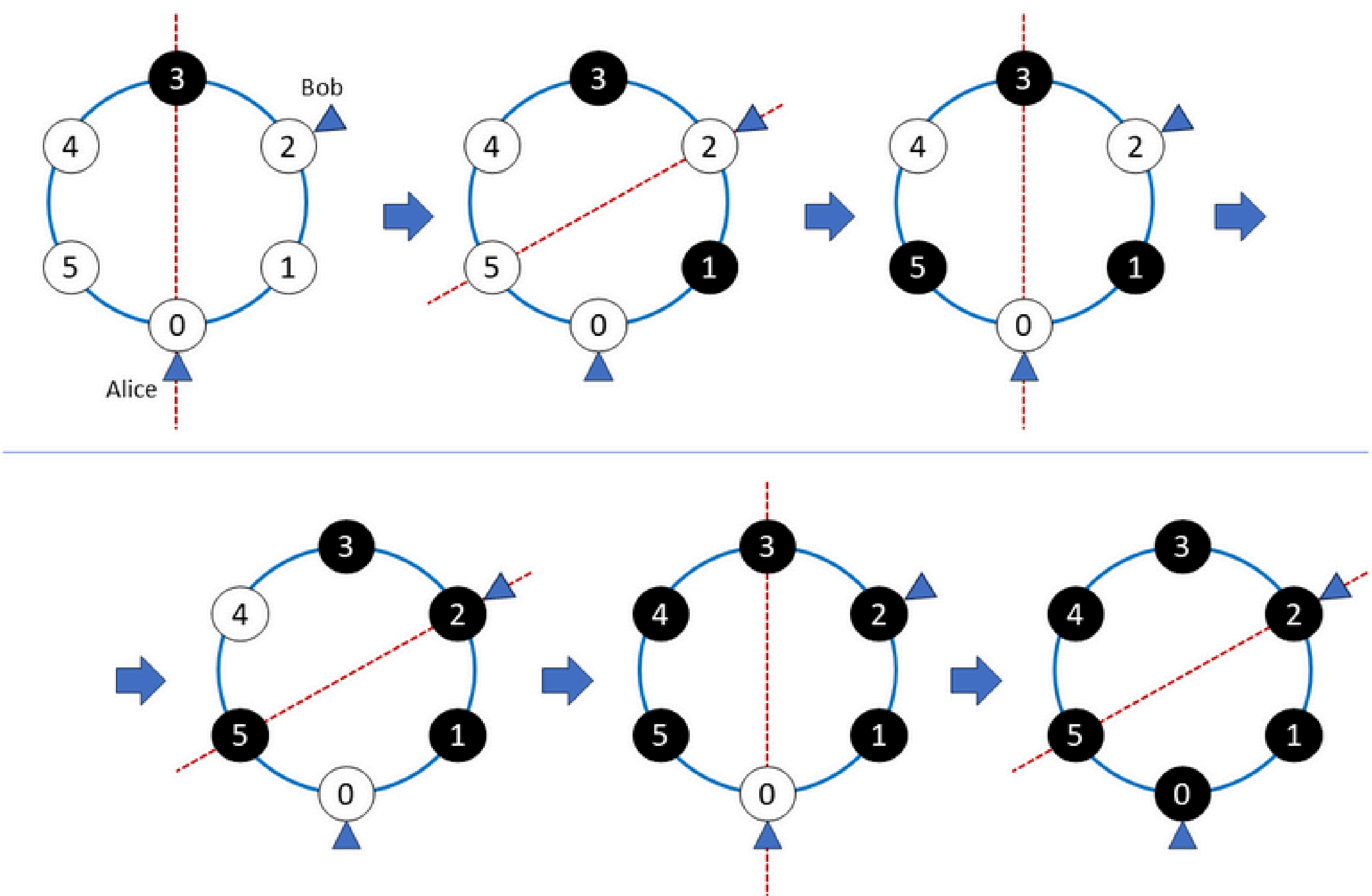
```
4
6 2
6 3
6 1
200000 100000
```

Sample Output 1

```
Yes
No
Yes
No
```

For $N = 6$ and $K = 2$, all points can be colored black by, for example, performing operations in the following order:

1. Alice colors point 3 black.
2. Bob colors point 1 black.
3. Alice colors point 5 black.
4. Bob colors point 2 black.
5. Alice colors point 4 black.
6. Bob colors point 0 black.



For $N = 6$ and $K = 3$, below is one possible progression. Actually, no matter what they do, they cannot color all points black.

1. Alice colors point 3 black.
2. Bob colors point 0 black.
3. Alice cannot color any point black so that the coloring will be symmetric with respect to her line, so she cannot perform the operation.

C - Honest or Liar or Confused

Time Limit: 2 sec / Memory Limit: 1024 MiB

Score : 700 points

Problem Statement

There is a village with N villagers numbered from 1 to N. Each villager is honest or a liar. Additionally, some villagers are confused.

You have obtained M testimonies from the villagers. Each testimony is given by A_i, B_i, C_i for $i = 1, 2, \ldots, M$, representing:

- If $C_i = 0$, villager A_i testified that villager B_i is honest.
- If $C_i = 1$, villager A_i testified that villager B_i is a liar.

All villagers know whether every other villager is honest or a liar, and you know that they made their testimonies to you according to the following rules:

- An honest villager who is not confused always tells the truth.
- A liar who is not confused always tells lies.
- A confused honest villager always tells lies.
- A confused liar always tells the truth.

In other words, if they are not confused, honest villagers always tell the truth, and liars always tell lies, but if they are confused, it is reversed.

You have decided to guess the set of villagers who are confused. Given a choice of villagers who are confused, whether the set of testimonies "contradicts" or not is determined. Here, a set of testimonies is said to contradict if, no matter how you assign honest or liar statuses to the villagers, there is at least one testimony that violates the villagers' testimony rules.

Find a set of confused villagers such that the given set of testimonies does not contradict. If no such set of confused villagers exists, indicate that fact.

Constraints

- $2 \leq N \leq 2 \times 10^5$
- $0 \leq M \leq \min\{2 \times 10^5, N(N-1)\}$
- $1 \leq A_i, B_i \leq N, A_i \neq B_i$
- $A_i \neq A_j$ or $B_i \neq B_j$ for $i \neq j$.
- $C_i = 0$ or 1.
- All input values are integers.

Input

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

$$
\begin{aligned}
&N \quad M \\
&A_1 \quad B_1 \quad C_1 \\
&A_2 \quad B_2 \quad C_2 \\
&\vdots \\
&A_M \quad B_M \quad C_M
\end{aligned}
$$

Output

If there exists a set of confused villagers such that the given set of testimonies does not contradict, print a string of length N representing the set of confused villagers. In this string, the i-th character should be `1` if villager i is confused, and `0` otherwise.

If no such set of confused villagers exists, print `-1`.

Sample Input 1

```
3 3
1 2 1
1 3 0
2 3 0
```

Sample Output 1

```
010
```

Suppose villager 1 is an honest villager who is not confused, villager 2 is a confused liar, and villager 3 is an honest villager who is not confused.

In this case, villager 1 correctly testifies that villager 2 is a liar and villager 3 is honest. Also, villager 2, who is a liar but confused, tells the truth and testifies that villager 3 is honest.

Therefore, all given testimonies are consistent with the villagers' testimony rules, so `010`, indicating that only villager 2 is confused, is one valid output.

Sample Input 2

```
3 6
1 2 1
1 3 0
2 1 1
2 3 0
3 1 1
3 2 0
```

Sample Output 2

```
-1
```

Suppose villagers 2 and 3 are confused.

In this case, there are $2^3 = 8$ possible combinations for whether each villager is honest or a liar. Among them, for example, if villager 1 is an honest villager who is not confused, villager 2 is a confused liar, and villager 3 is a confused honest villager, then according to the rules, villager 2 should tell the truth, but they falsely testify that villager 1 is a liar.

You can confirm that also in other combinations, there will be some testimonies that violate the rules.

Therefore, if villagers 2 and 3 are confused, the given set of testimonies contradicts.

In fact, in this test case, no matter which villagers are confused, the given set of testimonies contradicts.

Sample Input 3

```
3 0
```

Sample Output 3

```
000
```

There may be any number of confused villagers, possibly zero or all.

D - Mirror and Order

Time Limit: 2 sec / Memory Limit: 1024 MiB

Score : 1000 points

Problem Statement

You are going to create N sequences of length 3, satisfying the following conditions.

- For each of $k = 1, 2, 3$, the following holds:
  - Among the k-th elements of the sequences, each integer from 1 through N appears exactly once.

For this sequence of sequences, define sequences $a = (a_1, a_2, \ldots, a_N)$ and $b = (b_1, b_2, \ldots, b_N)$ as follows.

- Let s_i be the i-th sequence, and let t_i be the reverse of the i-th sequence. When all of these are sorted in lexicographical order, s_i comes a_i-th, and t_i comes b_i-th.
- Here, if there are identical sequences among the $2N$ sequences, a and b are not defined.

Therefore, if a and b are defined, each integer from 1 through $2N$ appears exactly once in the concatenation of a and b.

You are given sequences A and B of length N, where each element of A is an integer between 1 and $2N$, and each element of B is either an integer between 1 and $2N$ or -1. Also, in the concatenation of A and B, each integer other than -1 appears at most once.

How many pairs of sequences a, b are there such that a and b are defined and the following holds for each integer i from 1 through N?

- $a_i = A_i$.
- $b_i = B_i$ if $B_i \neq -1$.

Find the count modulo 998244353.

Constraints

- $2 \leq N \leq 3000$
- $1 \leq A_i \leq 2N$
- $1 \leq B_i \leq 2N$ or $B_i = -1$.
- In the concatenation of A and B, each integer other than -1 appears at most once. That is,
  - $A_i \neq A_j$ if $i \neq j$.
  - $B_i \neq B_j$ if $i \neq j$ and $B_i, B_j \neq -1$.
  - $A_i \neq B_j$.
- All input values are integers.

Input

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

$$
\begin{array}{l}
N \\
A_1 \quad A_2 \quad \ldots \quad A_N \\
B_1 \quad B_2 \quad \ldots \quad B_N
\end{array}
$$

Output

Print the count modulo 998244353.

Sample Input 1

```
3
2 3 6
-1 1 -1
```

Sample Output 1

```
1
```

For example, consider creating the following three sequences:

1. $(1, 2, 3)$
2. $(2, 1, 1)$
3. $(3, 3, 2)$

In this case, when sorting s_i and t_i lexicographically, we have:

> $t_2 = (1, 1, 2) < s_1 = (1, 2, 3) < s_2 = (2, 1, 1) < t_3 = (2, 3, 3) < t_1 = (3, 2, 1) < s_3 = (3, 3, 2)$

Thus, $(a_1, a_2, a_3, b_1, b_2, b_3) = (2, 3, 6, 5, 1, 4)$. Here, a matches the given A, and the second element of b also matches that of B, so this is one pair of sequences a, b satisfying the conditions.

On the other hand, if we create the following three sequences, s_1 and t_1 become identical, so a and b are not defined.

1. $(1, 2, 1)$
2. $(2, 1, 3)$
3. $(3, 3, 2)$

In fact, $a = (2, 3, 6), b = (5, 1, 4)$ is the only pair of sequences satisfying the conditions.

Sample Input 2

```
15
5 16 1 12 30 20 4 13 9 8 24 21 26 28 17
-1 -1 6 -1 -1 -1 -1 -1 -1 -1 -1 29 -1 -1 -1
```

Sample Output 2

```
758094847
```

Print the count modulo 998244353.

E - Gravity Sort

Time Limit: 2 sec / Memory Limit: 1024 MiB

Score : 1000 points

Problem Statement

There are $2N$ cells numbered from 1 to $2N$ arranged vertically in a column with cell 1 at the top. Each cell contains one ball. The weight of the ball in cell i at time $t = 0$ is m_i for $i = 1, 2, \ldots, N$, and 0 for $i = N+1, N+2, \ldots, 2N$. Here, $(m_1, m_2, \ldots, m_N)$ is a permutation of the integers from 1 to N.

In the following, we will refer to the ball of weight i as ball i, and the cell number where each ball is located as the position of the ball.

From time $t = 0$ onwards, every time the time advances by 1, the heavier balls sink downward, and the lighter balls rise upward.

Formally, the positions of each ball at time $t = t_0 + 1$ are determined from their positions at time $t = t_0$ by the following procedure.

- First, for $i = N, N-1, \ldots, 2, 1$ in this order, perform the following operation.
  - If the position of ball i at $t = t_0 + 1$ has already been determined:
    - **Do nothing.**
  - If the position of ball i at $t = t_0 + 1$ has not been determined:
    - If there exists a cell immediately below ball i at $t = t_0$, and the ball in that cell (call it ball j) is lighter than ball i, **set the positions of balls i and j at $t = t_0 + 1$ to be swapped from their positions at $t = t_0$.**
    - Otherwise, **set the position of ball i at $t = t_0 + 1$ to be the same as at $t = t_0$.**
- Next, for all balls of weight 0 whose positions at $t = t_0 + 1$ have not been determined at this point, **set their positions at $t = t_0 + 1$ to be the same as at $t = t_0$.**

It can be shown that at some time, the balls will be arranged from top to bottom in ascending order of weight, and their positions will no longer change. Find the time when this state is reached.

Constraints

- $1 \le N \le 2 \times 10^5$
- $1 \le m_i \le N$
- $m_i \neq m_j$ for $i \neq j$.
- All input values are integers.

Input

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

$$
\begin{array}{l}
N \\
m_1 \quad m_2 \quad \ldots \quad m_N
\end{array}
$$

Output

Print the answer as an integer.

Sample Input 1

```
3
2 3 1
```

Sample Output 1

```
6
```

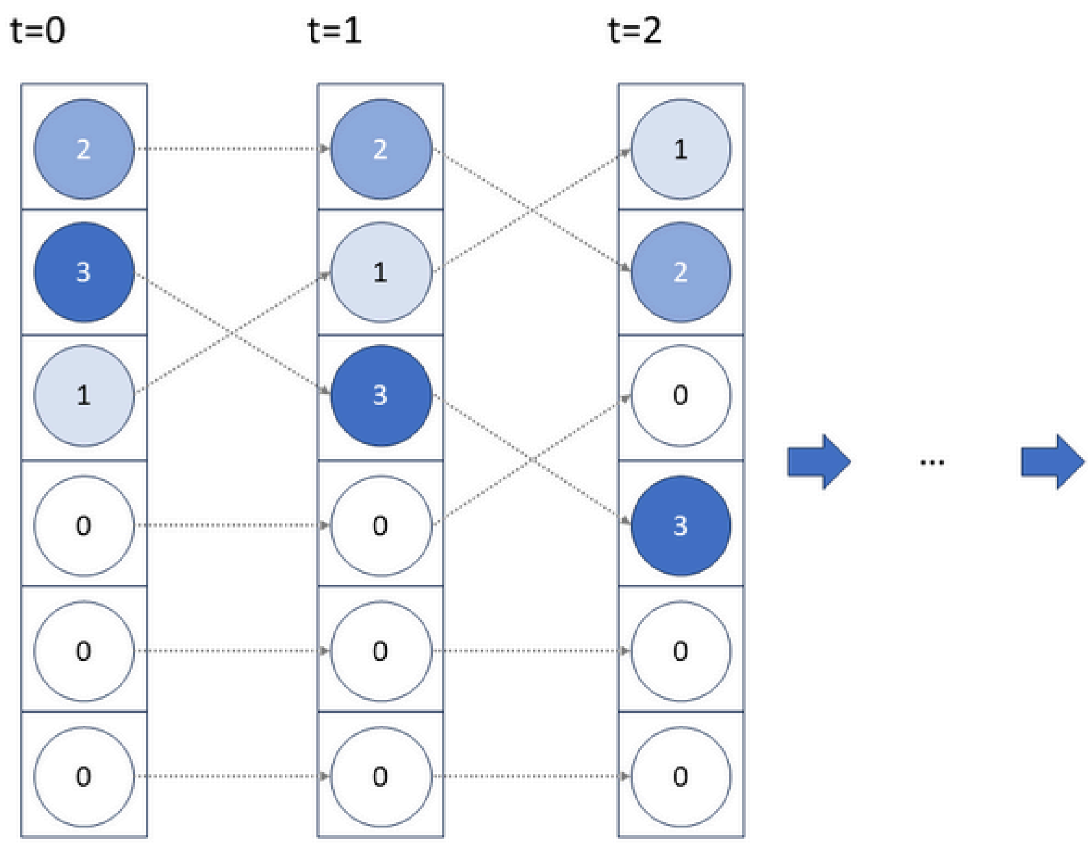
The movements from time $t = 0$ to $t = 1$ are determined as follows. (Refer to the diagram below if necessary.)

> 1. For ball 3, its position at $t = 1$ has not yet been determined. The cell immediately below it contains ball 1, and ball 3 is heavier, so swap their positions for $t = 1$. That is, set the position of ball 3 to cell 3, and ball 1 to cell 2.
> 2. For ball 2, its position at $t = 1$ has not yet been determined. The cell immediately below it contains ball 3, which is heavier than ball 2, so set the position of ball 2 at $t = 1$ to be the same as at $t = 0$.
> 3. For ball 1, its position at $t = 1$ has already been determined in the earlier step.
> 4. For balls of weight 0, none of their positions at $t = 1$ have been determined. Set their positions at $t = 1$ to be the same as at $t = 0$.

Next, the movements from time $t = 1$ to $t = 2$ are determined as follows.

> 1. For ball 3, its position at $t = 2$ has not yet been determined. The cell immediately below it contains ball 0, and ball 3 is heavier, so swap their positions for $t = 2$. That is, set the position of ball 3 to cell 4, and ball 0 (the one that was below ball 3) to cell 3.
> 2. For ball 2, its position at $t = 2$ has not yet been determined. The cell immediately below it contains ball 1, and ball 2 is heavier, so swap their positions for $t = 2$. That is, set the position of ball 2 to cell 2, and ball 1 to cell 1.
> 3. For ball 1, its position at $t = 2$ has already been determined in the earlier step.
> 4. For balls of weight 0, the one that was at cell 4 at $t = 1$ has already had its position at $t = 2$ determined in the earlier step. For the others, set their positions at $t = 2$ to be the same as at $t = 1$.

Continuing to determine the positions of the balls in this way, at time $t = 6$, the balls will be arranged from top to bottom as balls $0, 0, 0, 1, 2, 3$, and their positions will no longer change.



Sample Input 2

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

Sample Output 2

```
9
```

Sample Input 3

```
1
1
```

Sample Output 3

```
1
```