A. Was there an Array?

```
Input file:     standard input
Output file:    standard output
Time limit:     2 seconds
Memory limit:   512 megabytes
```

For an array of integers $a_1, a_2, \ldots, a_n$, we define its **equality characteristic** as the array $b_2, b_3, \ldots, b_{n-1}$, where $b_i = 1$ if the i-th element of the array a is equal to both of its neighbors, and $b_i = 0$ if the i-th element of the array a is not equal to at least one of its neighbors.

For example, for the array $[1, 2, 2, 2, 3, 3, 4, 4, 4, 4]$, the equality characteristic will be $[0, 1, 0, 0, 0, 0, 1, 1]$.

You are given the array $b_2, b_3, \ldots, b_{n-1}$. Your task is to determine whether there exists such an array a for which the given array is the equality characteristic.

Input

The first line contains one integer t ($1 \le t \le 1000$) — the number of test cases.

Each test case consists of two lines:

- the first line contains one integer n ($3 \le n \le 100$);
- the second line contains $n - 2$ integers $b_2, b_3, \ldots, b_{n-1}$ ($0 \le b_i \le 1$).

Output

For each test case, output YES if the array a exists, or NO if such an array does not exist. Each letter can be printed in any case.

Standard Input	Standard Output
3 10 0 1 0 0 0 0 1 1 3 1 10 0 1 0 1 1 0 0 1	YES YES NO

Note

In the first example, the array $a = [1, 2, 2, 2, 3, 3, 4, 4, 4, 4]$ is suitable.

In the second example, the array $a = [7, 7, 7]$ is suitable.

,

B. Set of Strangers

Input file: standard input
Output file: standard output
Time limit: 2 seconds
Memory limit: 256 megabytes

You are given a table of n rows and m columns. Initially, the cell at the i-th row and the j-th column has color $a_{i,j}$.

Let's say that two cells are *strangers* if they **don't** share a side. Strangers are allowed to touch with corners.

Let's say that the set of cells is a *set of strangers* if all pairs of cells in the set are strangers. Sets with no more than one cell are sets of strangers by definition.

In one step, you can choose any set of strangers **such that all cells in it have the same color** and paint all of them in some other color. You can choose the resulting color.

What is the minimum number of steps you need to make the whole table the same color?

Input

The first line contains a single integer t ($1 \le t \le 10^4$) — the number of test cases. Next, t cases follow.

The first line of each test case contains two integers n and m ($1 \le n \le m \le 700$) — the number of rows and columns in the table.

The next n lines contain the colors of cells in the corresponding row $a_{i,1}, \ldots, a_{i,m}$ ($1 \le a_{i,j} \le nm$).

It's guaranteed that the total sum of nm doesn't exceed $5 \cdot 10^5$ over all test cases.

Output

For each test case, print one integer — the minimum number of steps to paint all cells of the table the same color.

Standard Input	Standard Output
4 1 1 1 3 3 1 2 1 2 3 2 1 3 1 1 6 5 4 5 4 4 5 3 4 1 4 2 2 1 4 3 5 6 6 3 5	0 2 1 10

Note

In the first test case, the table is painted in one color from the start.

In the second test case, you can, for example, choose all cells with color 1 and paint them in 3. Then choose all cells with color 2 and also paint them in 3.

In the third test case, you can choose all cells with color 5 and paint them in color 4.

C. Beautiful Sequence

```
Input file:     standard input
Output file:    standard output
Time limit:     2 seconds
Memory limit:   256 megabytes
```

Let's call an integer sequence **beautiful** if the following conditions hold:

- its length is at least 3;
- for every element except the first one, there is an element to the left less than it;
- for every element except the last one, there is an element to the right larger than it;

For example, $[1, 4, 2, 4, 7]$ and $[1, 2, 4, 8]$ are beautiful, but $[1, 2]$, $[2, 2, 4]$, and $[1, 3, 5, 3]$ are not.

Recall that a subsequence is a sequence that can be obtained from another sequence by removing some elements without changing the order of the remaining elements.

You are given an integer array a of size n, where **every element is from 1 to 3**. Your task is to calculate the number of beautiful subsequences of the array a. Since the answer might be large, print it modulo 998244353.

Input

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

The first line of each test case contains a single integer n ($3 \le n \le 2 \cdot 10^5$).

The second line contains n integers $a_1, a_2, \dots, a_n$ ($1 \le a_i \le 3$).

Additional constraint on the input: the sum of n over all test cases doesn't exceed $2 \cdot 10^5$.

Output

For each test case, print a single integer — the number of beautiful subsequences of the array a, taken modulo 998244353.

Standard Input	Standard Output
4 7 3 2 1 2 2 1 3 4 3 1 2 2 3 1 2 3 9 1 2 3 2 1 3 2 2 3	3 0 1 22

Note

In the first test case of the example, the following subsequences are beautiful:

- $[a_3, a_4, a_7]$;
- $[a_3, a_5, a_7]$;
- $[a_3, a_4, a_5, a_7]$.

D. Palindrome Shuffle

```
Input file:     standard input
Output file:    standard output
Time limit:     2 seconds
Memory limit:   256 megabytes
```

You are given a string s consisting of lowercase Latin letters.

You can perform the following operation with the string s: choose a contiguous substring (possibly empty) of s and shuffle it (reorder the characters in the substring as you wish).

Recall that a palindrome is a string that reads the same way from the first character to the last and from the last character to the first. For example, the strings a, bab, acca, bcabcbacb are palindromes, but the strings ab, abbbaa, cccb are not.

Your task is to determine the minimum possible length of the substring on which the aforementioned operation must be performed in order to convert the given string s into a palindrome.

Input

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

The only line of each test case contains a string s ($2 \le |s| \le 2 \cdot 10^5$), consisting of lowercase Latin letters.

Additional constraints on the input:

- the string s has an even length;
- the string s can always be converted to a palindrome;
- the sum of lengths of s over all test cases doesn't exceed $2 \cdot 10^5$.

Output

For each test case, print a single integer — the minimum possible length of the substring on which the aforementioned operation must be performed in order to convert the given string s into a palindrome.

Standard Input	Standard Output
4 baba cc ddaa acbacddacbca	2 0 3 2

Note

In the first example, you can perform the operation as follows: $\mathtt{ba}\underline{\mathtt{ba}} \to \mathtt{ba}\underline{\mathtt{ab}}$.

In the second example, the string is already a palindrome, so we can shuffle an empty substring.

In the third example, you can perform the operation as follows: $\underline{\mathtt{dda}}\mathtt{a} \to \underline{\mathtt{add}}\mathtt{a}$.

In the fourth example, you can perform the operation as follows: $\mathtt{acb}\underline{\mathtt{ac}}\mathtt{ddacbca} \to \mathtt{acb}\underline{\mathtt{ca}}\mathtt{ddacbca}$.

,

E. A, B, AB and BA

```
Input file:     standard input
Output file:    standard output
Time limit:     2 seconds
Memory limit:   256 megabytes
```

You are given a string s consisting of characters A and B.

Your task is to split it into blocks of length 1 and 2 in such a way that

- there are no more than a strings equal to "A";
- there are no more than b strings equal to "B";
- there are no more than ab strings "AB";
- there are no more than ba strings "BA";

Strings "AA" and "BB" are prohibited. Each character of the initial string s should belong to exactly one block.

Input

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

The first line of each test case contains a single string s $(1 \le |s| \le 5 \cdot 10^5)$ consisting only of characters A and/or B.

The second line of each test case contains four integers a, b, ab, and ba $(0 \le a, b, ab, ba \le 5 \cdot 10^5)$ — the maximum allowed number of strings "A", "B", "AB", and "BA" correspondingly.

It's guaranteed that the total length of s doesn't exceed $5 \cdot 10^5$ over all test cases.

Output

For each test case, print YES if it's possible to split string s. Otherwise, print NO.

Standard Input	Standard Output
7 A 0 0 10 10 B 0 1 0 0 ABA 0 0 1 1 ABBABAAB 5 5 0 0 ABABBAABBAAB 1 1 2 3 ABBBBAB 0 3 2 0 BAABBA 1 3 2 0	NO YES NO YES YES YES NO

Note

In the third test case, all possible splits are: A | B | A, AB | A or A | BA. All of them have at least one "A".

In the fourth test case, one of the possible splits is the following: A | B | B | A | B | A | A | B.

In the fifth test case, one of the possible splits is the following: A | BA | B | BA | AB | BA | AB.

,

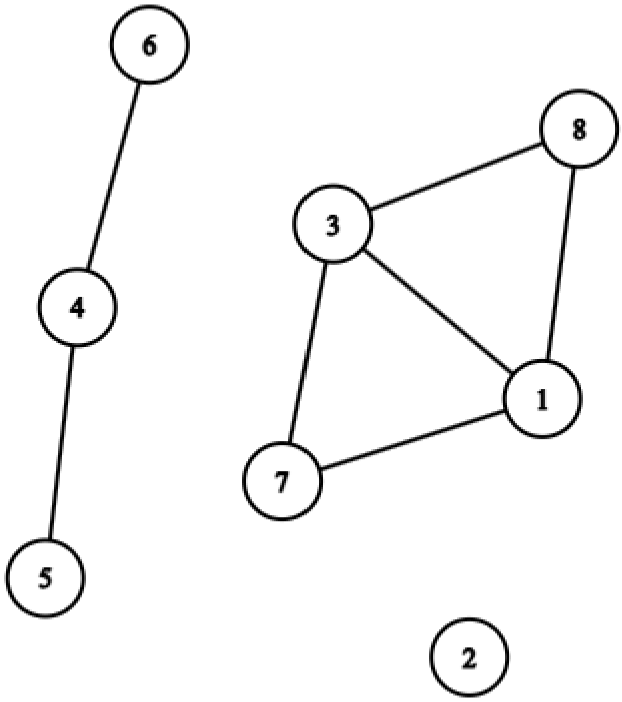
F. Graph Inclusion

```
Input file:     standard input
Output file:    standard output
Time limit:     5 seconds
Memory limit:   512 megabytes
```

A connected component of an undirected graph is defined as a set of vertices S of this graph such that:

- for every pair of vertices (u, v) in S, there exists a path between vertices u and v;
- there is no vertex outside S that has a path to a vertex within S.

For example, the graph in the picture below has three components: $\{1, 3, 7, 8\}$, $\{2\}$, $\{4, 5, 6\}$.



We say that graph A *includes* graph B if every component of graph B is a subset of some component of graph A.

You are given two graphs, A and B, both consisting of n vertices numbered from 1 to n. Initially, there are no edges in the graphs. You must process queries of two types:

- add an edge to one of the graphs;
- remove an edge from one of the graphs.

After each query, you have to calculate the minimum number of edges that have to be added to A so that A includes B, and print it. Note that you don't actually add these edges, you just calculate their number.

Input

The first line contains two integers n and q ($2 \le n \le 4 \cdot 10^5$; $1 \le q \le 4 \cdot 10^5$) — the number of vertices and queries, respectively.

Next, there are q lines, where the i-th line describes the i-th query. The description of the query begins with the character c_i (either A or B) — the graph to which the query should be applied. Then, two integers x_i and

y_i follow ($1 \le x_i, y_i \le n$; $x_i \ne y_i$). If there is an edge (x_i, y_i) in the corresponding graph, it should be removed; otherwise, it should be added to that graph.

Output

For each query, print one integer — the minimum number of edges that you have to add to the graph A so that it includes B.

Standard Input	Standard Output
6 9 A 2 3 B 1 3 A 2 1 A 3 2 B 5 6 A 6 5 A 3 4 A 4 2 A 4 3	0 1 0 1 2 1 1 0 1