A. Find Array

time limit per test: 1 second
memory limit per test: 256 megabytes
input: standard input
output: standard output

Given n, find any array $a_1, a_2, \ldots, a_n$ of integers such that all of the following conditions hold:

- $1 \le a_i \le 10^9$ for every i from 1 to n.
- $a_1 < a_2 < \ldots < a_n$
- For every i from 2 to n, a_i isn't divisible by a_{i-1}

It can be shown that such an array always exists under the constraints of the problem.

Input

The first line contains the number of test cases t ($1 \le t \le 100$). Description of the test cases follows.

The only line of each test case contains a single integer n ($1 \le n \le 1000$).

It is guaranteed that the sum of n over all test cases does not exceed 10^4.

Output

For each test case print n integers $a_1, a_2, \ldots, a_n$ — the array you found. If there are multiple arrays satisfying all the conditions, print any of them.

Example

input

```
3
1
2
7
```

output

```
1
2 3
111 1111 11111 111111 1111111 11111111 111111111
```

Note

In the first test case, array $[1]$ satisfies all the conditions.

In the second test case, array $[2, 3]$ satisfies all the conditions, as $2 < 3$ and 3 is not divisible by 2.

In the third test case, array $[111, 1111, 11111, 111111, 1111111, 11111111, 111111111]$ satisfies all the conditions, as it's increasing and a_i isn't divisible by a_{i-1} for any i from 2 to 7.

B. Build the Permutation

time limit per test: 1 second
memory limit per test: 256 megabytes
input: standard input
output: standard output

You are given three integers n, a, b. Determine if there exists a permutation $p_1, p_2, \ldots, p_n$ of integers from 1 to n, such that:

- There are exactly a integers i with $2 \le i \le n-1$ such that $p_{i-1} < p_i > p_{i+1}$ (in other words, there are exactly a local maximums).
- There are exactly b integers i with $2 \le i \le n-1$ such that $p_{i-1} > p_i < p_{i+1}$ (in other words, there are exactly b local minimums).

If such permutations exist, find any such permutation.

Input

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

The only line of each test case contains three integers n, a and b ($2 \le n \le 10^5$, $0 \le a, b \le n$).

The sum of n over all test cases doesn't exceed 10^5.

Output

For each test case, if there is no permutation with the requested properties, output -1.

Otherwise, print the permutation that you are found. If there are several such permutations, you may print any of them.

Example

input

```
3
4 1 1
6 1 2
6 4 0
```

output

```
1 3 2 4
4 2 3 1 5 6
-1
```

Note

In the first test case, one example of such permutations is $[1, 3, 2, 4]$. In it $p_1 < p_2 > p_3$, and 2 is the only such index, and $p_2 > p_3 < p_4$, and 3 the only such index.

One can show that there is no such permutation for the third test case.

C. Game Master

time limit per test: 1 second
memory limit per test: 256 megabytes
input: standard input
output: standard output

n players are playing a game.

There are two different maps in the game. For each player, we know his strength on each map. When two players fight on a specific map, the player with higher strength on that map always wins. No two players have the same strength on the same map.

You are the game master and want to organize a tournament. There will be a total of $n-1$ battles. While there is more than one player in the tournament, choose any map and any two remaining players to fight on it. The player who loses will be eliminated from the tournament.

In the end, exactly one player will remain, and he is declared the winner of the tournament. For each player determine if he can win the tournament.

Input

The first line contains a single integer t ($1 \le t \le 100$) — the number of test cases. The description of test cases follows.

The first line of each test case contains a single integer n ($1 \le n \le 10^5$) — the number of players.

The second line of each test case contains n integers $a_1, a_2, \dots, a_n$ ($1 \le a_i \le 10^9$, $a_i \neq a_j$ for $i \neq j$), where a_i is the strength of the i-th player on the first map.

The third line of each test case contains n integers $b_1, b_2, \dots, b_n$ ($1 \le b_i \le 10^9$, $b_i \neq b_j$ for $i \neq j$), where b_i is the strength of the i-th player on the second map.

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

Output

For each test case print a string of length n. i-th character should be "1" if the i-th player can win the tournament, or "0" otherwise.

Example

input

```
3
4
1 2 3 4
1 2 3 4
4
11 12 20 21
44 22 11 30
1
1000000000
1000000000
```

output

```
0001
1111
1
```

Note

In the first test case, the 4-th player will beat any other player on any game, so he will definitely win the tournament.

In the second test case, everyone can be a winner.

In the third test case, there is only one player. Clearly, he will win the tournament.

D. Dominoes

time limit per test: 1 second
memory limit per test: 256 megabytes
input: standard input
output: standard output

You are given n dominoes. Each domino has a left and a right cell. Each cell can be colored either black or white. Some cells are already colored, while some aren't yet.

The coloring is said to be **valid** if and only if it is possible to rearrange the dominoes in some order such that for each $1 \le i \le n$ the color of the right cell of the i-th domino is different from the color of the left cell of the $((i \bmod n) + 1)$-st domino.

Note that you can't rotate the dominoes, so the left cell always remains the left cell, and the right cell always remains the right cell.

Count the number of valid ways to color the yet uncolored cells of dominoes. Two ways are considered different if there is a cell that is colored white in one way and black in the other. In particular, colorings `BW WB` and `WB BW` different (and both invalid).

As this number can be very big, output it modulo $998\,244\,353$.

Input

The first line of the input contains a single integer n ($1 \le n \le 10^5$) — the number of dominoes.

The next n lines describe dominoes. Each line contains two characters which represent the left and the right cell. Character `B` means that the corresponding cell is black, character `W` means that the corresponding cell is white, and `?` means that the cell is yet to be colored.

Output

Print a single integer — the answer to the problem.

Examples

input

```
1
?W
```

output

```
1
```

input

```
2
??
W?
```

output

```
2
```

input

```
4
BB
??
W?
??
```

output

```
10
```

Note

In the first test case, there is only one domino, and we need the color of its right cell to be different from the color of its left cell. There is only one way to achieve this.

In the second test case, there are only 2 such colorings:

`BB WW` and `WB WB`.

E. The Cells on the Paper

time limit per test: 2 seconds
memory limit per test: 256 megabytes
input: standard input
output: standard output

On an endless checkered sheet of paper, n cells are chosen and colored in three colors, where n is divisible by 3. It turns out that there are exactly $\frac{n}{3}$ marked cells of each of three colors!

Find the largest such k that it's possible to choose $\frac{k}{3}$ cells of each color, remove all other marked cells, and then select three rectangles with sides parallel to the grid lines so that the following conditions hold:

- No two rectangles can intersect (but they can share a part of the boundary). In other words, the area of intersection of any two of these rectangles must be 0.
- The i-th rectangle contains all the chosen cells of the i-th color and no chosen cells of other colors, for $i = 1, 2, 3$.

Input

The first line of the input contains a single integer n — the number of the marked cells ($3 \le n \le 10^5$, n is divisible by 3).

The i-th of the following n lines contains three integers x_i, y_i, c_i ($|x_i|, |y_i| \le 10^9$; $1 \le c_i \le 3$), where (x_i, y_i) are the coordinates of the i-th marked cell and c_i is its color.

It's guaranteed that all cells (x_i, y_i) in the input are distinct, and that there are exactly $\frac{n}{3}$ cells of each color.

Output

Output a single integer k — the largest number of cells you can leave.

Examples

input

```
9
2 3 1
4 1 2
2 1 3
3 4 1
5 3 2
4 4 3
2 4 1
5 2 2
3 5 3
```

output

```
6
```

input

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

output

```
3
```

Note

In the first sample, it's possible to leave 6 cells with indexes $1, 5, 6, 7, 8, 9$.

In the second sample, it's possible to leave 3 cells with indexes $1, 2, 3$.

F. MEX counting

time limit per test: 4 seconds
memory limit per test: 256 megabytes
input: standard input
output: standard output

For an array c of nonnegative integers, $MEX(c)$ denotes the smallest nonnegative integer that doesn't appear in it. For example, $MEX([0, 1, 3]) = 2$, $MEX([42]) = 0$.

You are given integers n, k, and an array $[b_1, b_2, \ldots, b_n]$.

Find the number of arrays $[a_1, a_2, \ldots, a_n]$, for which the following conditions hold:

- $0 \le a_i \le n$ for each i for each i from 1 to n.
- $|MEX([a_1, a_2, \ldots, a_i]) - b_i| \le k$ for each i from 1 to n.

As this number can be very big, output it modulo $998\,244\,353$.

Input

The first line of the input contains two integers n, k ($1 \le n \le 2000$, $0 \le k \le 50$).

The second line of the input contains n integers $b_1, b_2, \ldots, b_n$ ($-k \le b_i \le n + k$) — elements of the array b.

Output

Output a single integer — the number of arrays which satisfy the conditions from the statement, modulo $998\,244\,353$.

Examples

input

```
4 0
0 0 0 0
```

output

```
256
```

input

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

output

```
431
```

input

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

output

```
509
```

input

```
5 2
0 0 2 2 0
```

output

```
6546
```

input

```
3 2
-2 0 4
```

output

```
11
```

G. Alphabetic Tree

time limit per test: 3 seconds
memory limit per test: 256 megabytes
input: standard input
output: standard output

You are given m strings and a tree on n nodes. Each edge has some letter written on it.

You have to answer q queries. Each query is described by 4 integers u, v, l and r. The answer to the query is the total number of occurrences of $str(u,v)$ in strings with indices from l to r. $str(u,v)$ is defined as the string that is made by concatenating letters written on the edges on the shortest path from u to v (in order that they are traversed).

Input

The first line of the input contains three integers n, m and q ($2 \le n \le 10^5$, $1 \le m, q \le 10^5$).

The i-th of the following $n-1$ lines contains two integers u_i, v_i and a lowercase Latin letter c_i ($1 \le u_i, v_i \le n$, $u_i \ne v_i$), denoting the edge between nodes u_i, v_i with a character c_i on it.

It's guaranteed that these edges form a tree.

The following m lines contain the strings consisting of lowercase Latin letters. The total length of those strings does not exceed 10^5.

Then q lines follow, each containing four integers u, v, l and r ($1 \le u, v \le n$, $u \ne v$, $1 \le l \le r \le m$), denoting the queries.

Output

For each query print a single integer — the answer to the query.

Examples

input

```
2 5 3
1 2 a
aab
abab
aaa
b
a
2 1 1 5
1 2 1 3
2 1 3 5
```

output

```
8
7
4
```

input

```
9 5 6
1 2 a
2 7 c
1 3 b
3 4 b
4 6 b
3 5 a
5 8 b
5 9 c
ababa
cabbb
bac
bbbac
abacaba
2 7 1 4
2 5 1 5
6 3 4 4
6 9 4 5
5 7 3 5
5 3 1 5
```

output

```
3
4
2
1
1
10
```