Problem A. Stop, Yesterday Please No More

After the great success in 2018, 2019, 2020 and 2021, Nanjing University of Aeronautics and Astronautics (NUAA) will host the *International Collegiate Programming Contest* (ICPC) Nanjing regional for the fifth time in a row.

Team ***Power of Two*** and team ***Three Hold Two*** won the champion title for Tsinghua University in 2018 and 2019. In 2020 and 2021, team ***Inverted Cross*** from Peking University won the back-to-back champion titles. They also won the second place in the 45th ICPC World Finals held in Dhaka, which is the best performance of East Continental teams in the past 6 years. We congratulate them and are very excited to see them coming back and compete in Nanjing 2022!

This year, there are around 500 teams participating in the contest. There are at most 35 gold medals, 70 silver medals and 105 bronze medals that will be awarded. We are looking forward to seeing participants' outstanding performance!

Although we can't gather in Nanjing this time (yet again) due to the pandemic, we should still be grateful for the hard work done by all staff and volunteers for this contest. Thank you all for your great contribution to this contest!

The 2018 ICPC Asia Nanjing Regional Contest

In the 2018 contest, problem K, ***Kangaroo Puzzle***, requires the contestants to construct an operation sequence for the game:

> The puzzle is a grid with n rows and m columns ($1 \le n, m \le 20$) and there are some (at least 2) kangaroos standing in the puzzle. The player's goal is to control them to get together. There are some walls in some cells and the kangaroos cannot enter the cells with walls. The other cells are empty. The kangaroos can move from an empty cell to an adjacent empty cell in four directions: up, down, left, and right.
>
> There is exactly one kangaroo in every empty cell in the beginning and the player can control the kangaroos by pressing the button U, D, L, R on the keyboard. The kangaroos will move simultaneously according to the button you press.
>
> The contestant needs to construct an operating sequence of at most 5×10^4 steps consisting of U, D, L, R only to achieve the goal.

In the 2020 contest, problem A, ***Ah, It's Yesterday Once More***, requires the contestants to construct an input map to hack the following code of the problem described before:

```
#include <bits/stdc++.h>
using namespace std;
string s = "UDLR";
int main()
{
  srand(time(NULL));
  for (int i = 1; i <= 50000; i++) putchar(s[rand() % 4]);
  return 0;
}
```

In the 2021 contest, problem A, ***Oops, It's Yesterday Twice More***, also requires the contestants to construct an operation sequence for the game:

This time, every cell in the grid stands exactly one kangaroo. You need to construct an operating sequence consisting only of characters 'U', 'D', 'L', and 'R'. After applying it, you must make sure every kangaroo will gather at the specific cell (a, b). The length of the operating sequence cannot exceed $3(n-1)$. As always, the kangaroos will move simultaneously according to the operation you command.

Now, in the 2022 contest, the kangaroo problem is back again! We don't know why problem setters are so obsessed with kangaroos but the problem is as follows:

You are given a grid with n rows and m columns. There is a hole in the cell on the i_h-th row and the j_h-th column. All other cells are empty and there is one kangaroo standing in each cell.

Similarly, the kangaroos are controlled by pressing the button U, D, L, R on the keyboard. All kangaroos will move simultaneously according to the button pressed. Specifically, for any kangaroo located in the cell on the i-th row and the j-th column, indicated by (i, j):

1. Button U: it will move to $(i-1, j)$.
2. Button D: it will move to $(i+1, j)$.
3. Button L: it will move to $(i, j-1)$.
4. Button R: it will move to $(i, j+1)$.

If a kangaroo steps onto the hole (that is, $i = i_h$ and $j = j_h$) or steps out of the grid, it will be removed from the grid.

The problem is that, the exact value of i_h and j_h is not known. You're only given an operating sequence consisting only of characters 'U', 'D', 'L', and 'R', and an integer k indicating that after applying the operating sequence, there are k kangaroos remaining on the grid.

Calculate the number of possible positions of the hole. That is, calculate the number of integer pairs (i_h, j_h) such that:

- $1 \le i_h \le n$, $1 \le j_h \le m$.
- The hole is located at (i_h, j_h).
- After applying the given operating sequence, the number of kangaroos remaining on the grid is exactly k.

Input

There are multiple test cases. The first line of the input contains an integer T indicating the number of test cases. For each test case:

The first line contains three integers n, m and k ($1 \le n, m \le 10^3$, $0 \le k < n \times m$) indicating the size of the grid and the number of kangaroos remaining on the grid after applying the operating sequence.

The second line contains a string $s_1 s_2 \cdots s_l$ ($s_i \in \{\text{`U'}, \text{`D'}, \text{`L'}, \text{`R'}\}$, $1 \le l \le 10^6$) indicating the operating sequence.

It's guaranteed that neither the sum of $n \times m$ nor the total length of the operating sequences of each test case will exceed 10^6.

Output

For each test case output one integer indicating the number of possible positions of the hole.

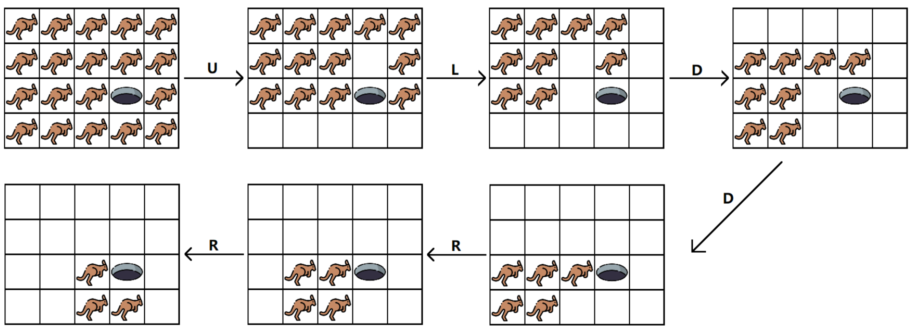
Example

standard input	standard output
3 4 5 3 ULDDRR 4 5 0 UUUUUUU 4 5 10 UUUUUUU	2 20 0

Note

For the first sample test case there are 2 possible positions for the hole.

The first possible position is $(3, 4)$.



The second possible position is $(4, 3)$.