

Problem A

Ants Sort

Time Limit: 2 seconds

Consider a line segment whose left and right endpoints are walls located at coordinates 0 and M , respectively. The positive direction is to the right.

There are N ants on the line segment, numbered $1, 2, \dots, N$ from left to right. The i -th ant is initially located at coordinate X_i ; thus, $0 < X_1 < X_2 < \dots < X_N < M$. It carries the ball numbered P_i and initially faces left if D_i is L, and right if D_i is R.

At time 0, all ants start moving in the direction they are facing at a speed of 1 unit per second.

When an ant collides with a wall, it immediately reverses its direction. When two ants collide, both immediately reverse direction. At that moment, you may choose whether or not to swap the balls they are carrying.

Your goal is to make the i -th ant carry the ball numbered i for every i ($1 \leq i \leq N$). It can be proven that this goal can always be achieved. Find the earliest time at which this can be achieved.

Input

The input consists of a single test case of the following format.

```
N M
X1 P1 D1
X2 P2 D2
⋮
XN PN DN
```

The first line contains two integers N and M ($2 \leq N \leq 2 \times 10^5$, $6 \leq M \leq 10^{15}$, and M is even), representing the number of ants and the coordinate of the right wall, respectively.

For each i ($1 \leq i \leq N$), the i -th of the following N lines contains two integers X_i and P_i , and a character D_i , representing the initial coordinate of the i -th ant, the number of the ball it is initially carrying, and its initial direction, respectively. The coordinates $X_1, X_2, \dots, X_N$ are even integers satisfying $0 < X_1 < X_2 < \dots < X_N < M$. The sequence $(P_1, P_2, \dots, P_N)$ is a permutation of $(1, 2, \dots, N)$ and is not equal to $(1, 2, \dots, N)$. Each D_i is either L or R.

Output

Print the earliest time at which this goal can be achieved. It is guaranteed that the answer is an integer.

Sample Input 1

```
3 10
2 3 R
4 1 R
8 2 L
```

Sample Output 1

```
7
```

Sample Input 2

```
2 943168
168 2 L
874 1 R
```

Sample Output 2

```
942815
```

Sample Input 3

```
9 168598874
2 2 L
4 9 R
8 4 L
16 3 L
32 1 R
64 6 R
128 8 R
256 5 R
512 7 L
```

Sample Output 3

```
168598856
```

Problem B

ABbreviation

Time Limit: 2 seconds

You are given a string S consisting of A and B.

You may perform the following operation on S any number of times, possibly zero.

- Choose an occurrence of AB as a contiguous substring of S , and replace these two characters with a single uppercase English letter (one of A through Z) of your choice. If the replacement letter is A or B, it may be used as part of AB in a later operation.

For example, if S is AAB, you may replace the last two characters AB with C, resulting in AC. If you replace them with B instead, the resulting string is AB, on which you may perform the operation again.

Find the number of distinct strings that can be obtained from S by performing the operation any number of times.

Since the answer may be large, print it modulo 998 244 353.

Input

The input consists of a single test case of the following format.

N
 S

The first line contains an integer N ($1 \leq N \leq 10^6$), representing the length of S .

The second line contains a string S of length N consisting of A and B.

Output

Print a single integer representing the number of distinct strings that can be obtained, modulo 998 244 353.

Sample Input 1

4 AABB	104
-----------	-----

Sample Input 2

1 A	1
--------	---

Sample Input 3

17 ABBABABAAABBAABBB	894459821
-------------------------	-----------

Problem C

Line Up!

Time Limit: 2 seconds

There are N people on a two-dimensional plane. The i -th person is initially located at (x_i, y_i) and can move freely in any direction at a speed of at most v_i units per second.

They all want to line up on a single straight line. Find the minimum time required for them to do so.

More precisely, find the minimum nonnegative real number T for which there exists a straight line such that every person can reach some point on it within T seconds.

The position and orientation of the line may be chosen freely. Multiple people may occupy the same point.

Input

The input consists of a single test case of the following format.

```
N
x1 y1 v1
x2 y2 v2
⋮
xN yN vN
```

The first line contains an integer N ($3 \leq N \leq 1\,000$), representing the number of people.

For each i ($1 \leq i \leq N$), the i -th of the following N lines contains three integers x_i , y_i , and v_i ($-10^9 \leq x_i, y_i \leq 10^9$, $1 \leq v_i \leq 10^5$). The i -th person is initially located at (x_i, y_i) and has a maximum speed of v_i units per second.

Output

Print a single real number representing the minimum time required for all N people to be located on a single straight line.

Your answer will be considered correct if its absolute or relative error does not exceed 10^{-7} .

Sample Input 1

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

Sample Output 1

```
0.471404520791
```

Sample Input 2

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

Sample Output 2

```
0.000000000000
```

Sample Input 3

```
26
226639011 714030491 52084
-433602126 -108920882 63933
529792924 -599056247 51073
447686200 -908930834 96957
752558066 -98596171 12331
-325499883 966751315 64309
507655993 356046254 24621
323698442 815958050 1579
-797812857 -247507527 45097
952346469 565751814 86072
-631841229 823378480 3226
-139573806 864389108 31195
-238947644 27719504 22660
-834033851 -831292724 26624
947551802 -473531913 65103
829291610 99663009 45079
-764217971 -751329486 28073
-687089797 -200100828 3458
-597429979 -713064249 94885
273582861 727423928 31031
-423665895 -244614618 46129
310263329 834575868 19397
556376001 -539503889 51869
452257555 672020980 48905
-244101763 708382053 51097
-724609724 489151554 38920
```

Sample Output 3

```
118687.975695169065
```

Problem D

Self-Destruct Sequence

Time Limit: 2 seconds

You are given a sequence of N positive integers $A = (A_1, A_2, \dots, A_N)$.

For each k ($1 \leq k \leq N$), solve the following problem:

Let B initially be the sequence $(A_1, A_2, \dots, A_k)$. At any point during the process, let $|B|$ denote the current length of B and let B_i denote its i -th element.

You may perform the following operation any number of times.

- Choose an index i ($1 \leq i \leq |B|$) satisfying $i + B_i - 1 \leq |B|$. Remove the elements $B_i, B_{i+1}, \dots, B_{i+B_i-1}$ from B . The remaining elements, in their original order, form the new sequence B .

Determine whether B can be made empty. If so, find the minimum and maximum possible numbers of operations required.

Input

The input consists of a single test case of the following format.

N
 $A_1 A_2 \dots A_N$

The first line contains an integer N ($1 \leq N \leq 5\,000$), representing the length of the sequence.

The second line contains N integers $A_1, A_2, \dots, A_N$ ($1 \leq A_i \leq N$).

Output

Print N lines.

For each k ($1 \leq k \leq N$), the k -th line should contain the answer for $B = (A_1, A_2, \dots, A_k)$. If it is impossible to make B empty, print -1. Otherwise, print two integers: the minimum and maximum possible numbers of operations required to make B empty, respectively.

Sample Input 1

5
3 4 2 1 1

Sample Output 1

-1
-1
1 1
2 2
2 3

Sample Input 2

6
2 1 3 2 1 2

Sample Output 2

-1
1 1
2 2
-1
2 3
3 4

Problem E

Box Tower II

Time Limit: 2 seconds

There are N cube-shaped boxes of the same size, numbered $1, 2, \dots, N$. Each face of each box is colored either black or white.

The colors of the six faces of the i -th box are denoted by u_i, d_i, f_i, b_i, l_i , and r_i , as shown in Figure E-1. Here, each of u_i, d_i, f_i, b_i, l_i , and r_i is either B or W, where B represents black and W represents white.

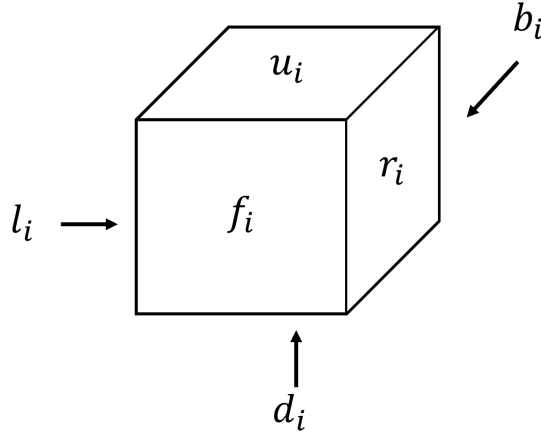


Figure E-1: Correspondence between the input and the faces of a box

You are given Q queries. For each q ($1 \leq q \leq Q$), the q -th query specifies two integers s_q and t_q .

Using all the boxes numbered $s_q, s_q+1, \dots, t_q$, you would like to make a single tower of boxes by rotating each box and then stacking the boxes vertically. After the orientation of each box is determined, all boxes are placed so that their top and bottom faces are horizontal and their bases are exactly aligned. Thus, each of the four sides of the tower consists of exactly one face from each box.

A tower is called *balanced* if, on each of its four sides, the number of black faces is equal to the number of white faces.

For each q ($1 \leq q \leq Q$), determine whether it is possible to make a balanced tower using all the boxes numbered $s_q, s_q + 1, \dots, t_q$.

Input

The input consists of a single test case of the following format.

```

N Q
u1d1f1b1l1r1
u2d2f2b2l2r2
⋮
uNdNfNbNlNrN
s1 t1
s2 t2
⋮
sQ tQ

```

The first line contains two integers N and Q ($1 \leq N \leq 2 \times 10^5, 1 \leq Q \leq 2 \times 10^5$), representing the

number of boxes and the number of queries, respectively.

For each i ($1 \leq i \leq N$), the i -th of the following N lines contains the six characters u_i, d_i, f_i, b_i, l_i , and r_i , in this order and without spaces. Each character is either B or W and represents the color of the corresponding face of the i -th box as shown in Figure E-1.

For each q ($1 \leq q \leq Q$), the q -th of the following Q lines contains two integers s_q and t_q ($1 \leq s_q \leq t_q \leq N$), representing the range of boxes used in the q -th query.

Output

Print Q lines.

For each q ($1 \leq q \leq Q$), print Yes on the q -th line if it is possible to make a balanced tower using all the boxes numbered $s_q, s_q + 1, \dots, t_q$, and print No otherwise.

Sample Input

```
4 5
BBWBBW
WWWBBB
BWBBWW
BBBBBB
1 2
2 3
3 4
1 4
2 2
```

Sample Output

```
Yes
Yes
No
Yes
No
```

Problem F vs Hydra

Time Limit: 2 seconds

A Hydra with N heads has appeared. The i -th head initially has H_i health points.

You may perform attacks any number of times, including zero. In each attack, choose one of the following $N + 1$ types:

- Attack i ($1 \leq i \leq N$): Decrease the health points of the i -th head by X .
- Attack $N + 1$: Decrease the health points of every head by Y .

Find the minimum number of attacks required to make the health points of every head at most 0.

Input

The input contains one or more test cases. The first line of the input contains an integer T ($1 \leq T \leq 100$), which is the number of test cases. The descriptions of the T test cases follow, each in the following format.

$$\begin{array}{l} N \ X \ Y \\ H_1 \ H_2 \ \dots \ H_N \end{array}$$

For each test case, the first line contains three integers N , X , and Y ($1 \leq N \leq 3 \times 10^5$, $1 \leq X, Y \leq 10^{18}$), representing the number of heads, the damage dealt by Attack i ($1 \leq i \leq N$), and the damage dealt by Attack $N + 1$, respectively.

The second line contains N integers $H_1, H_2, \dots, H_N$ ($1 \leq H_i \leq 10^{18}$), where H_i represents the initial health points of the i -th head.

The sum of N over all test cases does not exceed 3×10^5 .

Output

For each test case, print a single integer representing the minimum number of attacks required to make the health points of every head at most 0.

Sample Input

```
2
3 5 2
12 6 17
1 998244353 1000000007
1000000007
```

Sample Output

```
7
1
```

Problem G

Guess Palindrome

Time Limit: 2 seconds

Alice has a hidden string S of length N consisting only of the two characters a and b. Since she likes symmetric strings, it is known that S is a palindrome. Your task is to determine S by asking Alice the following type of query:

- You provide an arbitrary non-empty string T of length at most N consisting only of a and b.
- Alice responds with an integer representing the number of occurrences of T as a contiguous substring of S . Occurrences are counted even if they overlap.

However, to avoid upsetting Alice, you may ask at most $\lfloor N/2 \rfloor$ queries.

Write a program that determines the hidden string S within the given query limit.

Interaction

First, the length N of S ($2 \leq N \leq 100$) is given from standard input.

After reading the input, you may start asking queries. To ask a query with a string T , print the following format to standard output, where T must be a non-empty string of length at most N consisting only of a and b.

? T

In response to this query, an integer representing the number of occurrences of T as a contiguous substring of S is given from standard input.

Once you have determined S , print your answer to standard output in the following format.

! S

You may ask at most $\lfloor N/2 \rfloor$ queries and print the answer exactly once. After printing the answer, your program must terminate immediately.

If your program violates the specified format, exceeds the query or answer limit, or produces any extraneous output, your submission will be judged as a wrong answer. Note that you may need to flush the output buffer after each output. Note also that the judge for this problem is not adaptive. The judge determines S before the interaction begins.

Sample Interaction

Read	Write
9	
1	? aabb
1	? baa
4	? aa
	! bbaaaaabb

Problem H

Noise Cancelling

Time Limit: 3 seconds

A sequence X consisting of nonzero integers is called *noise-cancelling* if there exists a positive integer m such that its length is $2m$ and

$$x_i + x_{i+m} = 0$$

holds for every i ($1 \leq i \leq m$), where x_i denotes the i -th element of X . In other words, the second half of a noise-cancelling sequence is the elementwise negation of its first half, so the corresponding elements cancel each other out. In particular, no sequence of odd length is noise-cancelling.

For example, $(1, 2, -3, 4, -1, -2, 3, -4)$ is noise-cancelling.

You are given a sequence of N nonzero integers $(a_1, a_2, \dots, a_N)$. Find the number of pairs of integers (l, r) ($1 \leq l \leq r \leq N$) such that the contiguous subsequence $(a_l, a_{l+1}, \dots, a_r)$ is noise-cancelling.

Input

The input consists of a single test case of the following format.

N
 $a_1 \ a_2 \ \dots \ a_N$

The first line contains an integer N ($1 \leq N \leq 2 \times 10^5$), representing the length of the sequence.

The second line contains N nonzero integers $a_1, a_2, \dots, a_N$ ($1 \leq |a_i| \leq N$), representing the elements of the sequence.

Output

Print a single integer representing the number of pairs (l, r) such that $(a_l, a_{l+1}, \dots, a_r)$ is noise-cancelling.

Sample Input 1

9
1 -3 2 -2 -1 3 -2 2 -2

Sample Output 1

4

Sample Input 2

20
-1 -1 -1 1 1 1 -1 -1 1 1 -1 -1 -1 1 1 -1 1 -1 -1 1

Sample Output 2

26

Problem I

JAG City Marathon II

Time Limit: 2 seconds

Just A Grid City (JAG City) has N horizontal roads and N vertical roads. These roads form a square grid. The distance between any two neighboring parallel roads is 1. The horizontal roads are numbered from 1 to N from top to bottom, and the vertical roads are numbered from 1 to N from left to right. We call the intersection of horizontal road i and vertical road j (i, j) .

A marathon will be held on the roads of JAG City. To make the runners visit many parts of the city, the organizers have placed N checkpoints. You are given a permutation $P = (P_1, P_2, \dots, P_N)$ of $(1, 2, \dots, N)$. The checkpoints are placed at (i, P_i) ($i = 1, 2, \dots, N$).

The organizers want the course to be easy to follow. Therefore, the marathon course must satisfy all of the following conditions:

- The course is a closed route that uses only the roads, visits its starting point only at the start and at the end, and does not visit any other point more than once.
- The course uses **all** N horizontal roads and **all** N vertical roads, traveling a distance of at least 1 along every road.
- The course passes through every checkpoint and **turns at every checkpoint**.
- The course has exactly $2N$ turns.

In particular, the course **must not cross itself**, even at intersections where it does not turn. At each turn, the course switches from a horizontal road to a vertical road or vice versa.

A route is valid if it satisfies all the conditions above.

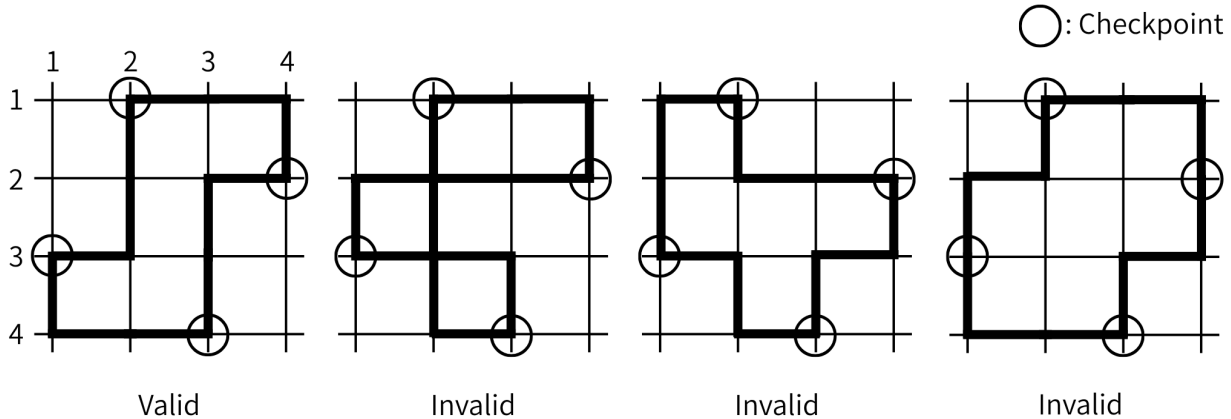


Figure I-1: Examples of valid and invalid marathon courses

Figure I-1 shows examples of valid and invalid routes for the sample input. The leftmost route is valid because it satisfies all the conditions. The second route is invalid because it crosses itself. The third route is invalid because it has more than $2N$ turns. The rightmost route is invalid because it does not turn at every checkpoint.

Output one marathon course that satisfies all the conditions. It is guaranteed that such a course exists under the given constraints.

Note that you **do not need to maximize or minimize** the total length of the course.

Input

The input consists of a single test case of the following format.

$$\begin{array}{c} N \\ P_1 \ P_2 \ \dots \ P_N \end{array}$$

The first line contains an integer N ($2 \leq N \leq 2 \times 10^5$), representing the number of roads in each direction.

The second line contains N distinct integers $P_1, P_2, \dots, P_N$ ($1 \leq P_i \leq N$), which form a permutation P .

Output

Output the $2N$ intersections at which the course turns, in the order in which they are visited along the course, in the following format:

$$\begin{array}{c} X_1 \ Y_1 \\ X_2 \ Y_2 \\ \vdots \\ X_{2N} \ Y_{2N} \end{array}$$

For each i ($1 \leq i \leq 2N$), the pair (X_i, Y_i) represents the intersection of horizontal road X_i and vertical road Y_i , where $1 \leq X_i, Y_i \leq N$.

If there are multiple valid routes, you may output any of them.

You may choose any turning intersection as (X_1, Y_1) and output the intersections in either direction along the course.

Sample Input	Sample Output
4 2 4 1 3	2 4 1 4 1 2 3 2 3 1 4 1 4 3 2 3

Problem J

Forbidden Word

Time Limit: 3 seconds

You are a highly skilled programmer working in the social media monitoring division of ICPC (International Company of Public Communication), a giant corporation. Your task is to write a program that determines whether a message scheduled to be posted on social media by the public relations department contains a predefined forbidden word.

In recent years, numerous social media controversies have been caused by unintended hidden messages that appear when a post is wrapped at a particular number of characters and the characters in each column are read vertically. Forbidden words that appear horizontally within a displayed row must also be avoided.

For example, consider the forbidden word `dk` and the message `abcdefghijklmn`. When the viewer's screen width is 10 characters, the message is displayed as shown in Figure J-1, and `dk` does not appear when the columns are read vertically. However, when the screen width is 7 characters, the message is displayed as shown in Figure J-2, and reading the fourth column from top to bottom yields `dk`.

abcdefghijkl
mn

Figure J-1: Screen width 10.

abcdefg
hijklmn

Figure J-2: Screen width 7.

You are given a string S of length N , representing the message to be posted, and a string T of length M , representing the forbidden word. Because social media users may view the message on screens of various widths, you must determine whether the message is at risk of causing a controversy for each possible screen width i ($1 \leq i \leq N$).

For a screen width of i , the string S is displayed with a line break after every i characters, starting from the beginning of the string. More precisely, the j -th character of S ($1 \leq j \leq N$) is placed in row $\lfloor (j-1)/i \rfloor + 1$ from the top and column $((j-1) \bmod i) + 1$ from the left. Note that the last row may contain fewer than i characters.

For a screen width of i , the message is considered *at risk of causing a controversy* if T occurs as a contiguous substring in the left-to-right reading of some row or in the top-to-bottom reading of some column.

Input

The input contains one or more test cases. The first line of the input contains an integer t ($1 \leq t \leq 100$), which is the number of test cases. The descriptions of the t test cases follow, each in the following format.

N M
 S
 T

The first line of each test case contains two integers N and M ($1 \leq M \leq N \leq 2 \times 10^5$), representing the lengths of the message and the forbidden word, respectively. The second line contains a string S of length N consisting of lowercase English letters. The third line contains a string T of length M consisting of lowercase English letters.

The sum of N over all test cases does not exceed 2×10^5 , and the sum of M over all test cases does not exceed 2×10^5 .

Output

For each test case, output a binary string of length N . For each i ($1 \leq i \leq N$), the i -th character of the string should be 1 if the message is at risk of causing a controversy for a screen width of i , and 0 otherwise.

Sample Input	Sample Output
5	00000010000000
14 2	101111
abcdefghijklmn	0101010100
dk	110111111111
6 3	11111111
banana	
ana	
10 2	
ababababab	
aa	
11 4	
mississippi	
issi	
8 3	
aaaaaaaa	
aaa	

Problem K

Two Friend Groups

Time Limit: 2 seconds

JAG Zoo keeps N monkeys. The monkeys are numbered from 1 to N .

The zoo staff have investigated the relationships among the monkeys. For each of M pairs of monkeys, they have determined whether the two monkeys like each other or not. Based on this information, you decide to divide the N monkeys into two non-empty cages so that they can live as peacefully as possible. For each of these pairs, the following conditions must be satisfied:

- Two monkeys that like each other must be placed in the same cage.
- Two monkeys that do not like each other must be placed in different cages.

Determine whether it is possible to divide the monkeys in this way.

Input

The input consists of a single test case of the following format.

```
 $N$   $M$ 
 $a_1$   $b_1$   $c_1$ 
 $a_2$   $b_2$   $c_2$ 
 $\vdots$ 
 $a_M$   $b_M$   $c_M$ 
```

The first line contains two integers N and M ($2 \leq N \leq 10^5$, $1 \leq M \leq 10^5$), representing the number of monkeys and the number of pairs, respectively.

Each of the following M lines contains two integers a_i and b_i ($1 \leq a_i, b_i \leq N$) and a character c_i , which is either o or x. If c_i is o, monkeys a_i and b_i like each other. If c_i is x, they do not like each other. It is guaranteed that $a_i \neq b_i$ and that no pair of monkeys appears more than once.

Output

If it is possible to divide the monkeys into two non-empty cages, print Yes; otherwise, print No.

Sample Input 1

```
5 8
2 3 x
2 5 x
1 3 x
3 4 o
1 5 x
4 5 o
3 5 o
1 4 x
```

Sample Output 1

```
Yes
```

Sample Input 2

```
2 1
1 2 o
```

Sample Output 2

```
No
```

Problem L

Treasure Hunting

Time Limit: 2 seconds

You are playing a treasure-hunting video game that takes place on a two-dimensional plane.

You have a string s of length m consisting only of R, U, and ?. First, you replace each occurrence of ? in s with either R or U.

After replacing all occurrences of ?, you start at $(0,0)$ and make m moves according to s . Let (x,y) denote your current location. On the i -th move, if the i -th character of s is R, your next location is $(x+1,y)$; if it is U, your next location is $(x,y+1)$.

There are n treasure boxes on the plane. The j -th treasure box is located at (a_j, b_j) , and if you reach this location, you can open the box to collect c_j coins. Multiple treasure boxes may be located at the same point. If you reach such a point, you can open all the treasure boxes there and collect the coins from each of them.

Find the maximum number of coins you can collect if you optimally replace each occurrence of ? in s .

Input

The input consists of a single test case of the following format.

```
m n
s
a1 b1 c1
a2 b2 c2
⋮
an bn cn
```

The first line contains two integers m and n . m is the total number of moves in this game ($1 \leq m \leq 3 \times 10^5$) and n is the number of treasure boxes ($1 \leq n \leq 3 \times 10^5$).

The second line contains a string s of length m . Each character of s is either R, U, or ?.

Each of the next n lines contains three integers a_j , b_j , and c_j . (a_j, b_j) is the location of the j -th treasure box ($a_j \geq 0$, $b_j \geq 0$, $1 \leq a_j + b_j \leq m$). c_j is the number of coins in the box ($1 \leq c_j \leq 10^9$).

Output

Output the maximum number of coins you can collect.

Sample Input 1

```
8 5
R?UU?U?R
1 2 3
2 3 5
1 6 1
3 5 4
5 3 10
```

Sample Output 1

```
12
```

Sample Input 2

```
10 2
RRRRUUUUURR
5 0 10
0 1 8
```

Sample Output 2

```
0
```

Sample Input 3

```
20 15
?RU?R?????U?R?UR???U
1 7 19
16 4 13
1 0 15
8 3 20
7 8 18
2 10 17
5 9 15
14 6 16
0 1 6
11 1 19
10 2 14
12 3 11
4 11 18
2 3 16
2 16 11
```

Sample Output 3

```
57
```

In Sample Input 1, one optimal replacement results in the string RUUURUUR. With this string, you can open the first, second, and fourth treasure boxes and collect 12 coins.