Mario 2

Problem Description

Imagine you are a video game developer. You are developing a game which requires the player to collect coins and cross hurdles. Let the character in your video game be called Mario. As Mario moves to collect coins and cross hurdles, the game keeps a count of relevant metrics. Write code to implement this flow.

Mario will run from left to right and jump from the ground in the air to collect coins or cross hurdles. The Game Screen will be provided as input in form of a matrix comprising of three characters viz {0, C and H}, where

0 - denotes empty space

C - denotes coins to be collected

H - denotes hurdles to be crossed

All coins are of the same type, whereas there are two types of hurdles - simple hurdle and ring hurdle. Simple hurdle is referred to as Hurdle hereafter.

A Hurdle always begins from the ground and a series of the letter H stacked vertically make up the height of the hurdle.

A Ring Hurdle on the other hand, has a hole in it i.e., between H characters there will be exactly one hole denoted by 0 character. This hole is big enough for Mario to jump through it to cross that hurdle.

Now, let us understand how this information is provided in the input

- The screen will be depicted in the input as a M * N matrix. The index of row and columns of this matrix begin from zero.
- The left bottom cell of this matrix is (0, 0). As we move right and up, the row and column indices increase
- Row zero is considered as Ground and anything above row zero is considered as Air
- Coins will always be in air, whereas hurdles will always manifest from the ground
- In order to jump, Mario needs to have a calorific budget. This jump can be for collecting coins or crossing hurdles
- For Jumping, one row worth of height, *JumpCalorie* worth of calories is required
- On obtaining one coin, Mario receives *CoinCalorie* worth of calories
- Game starts with Mario having an initial number of calories, I
- The game is over in one of the following conditions
    - Either Mario crosses the grid and reaches the other end or
    - If Mario falls short of calorific budget to either be able to collect coins or cross hurdles
- To collect coins Mario will jump vertically in the column where the coin is. Mario always jumps to the highest point where a coin is, on the screen. On his way down from that point, he grabs all coins lower in height in that column. Thus, one jump in one column is enough to fetch all coins in that column
- Mario cannot jump and collect the lowest coin. He must always aim for the highest coin in that column
- Mario never jumps unless he must collect coins or cross a hurdle
- When crossing a ring hurdle, the calories consumed in clearing it is *JumpCalorie* * height of the hole in the ring hurdle. Refer *Examples* section for better clarity

- Walking i.e., moving from one column to another
    - Consumes 1 calorie if, no coins are collected in the source column
    - Consumes 0 calories if, one or more coins are collected in the source column

Your task is to keep a track of the last column Mario has successfully crossed.

Consider the following as inputs along with a screen(grid) of size 5 * 10:

Initial calorie: 5

JumpCalorie: 1

CoinCalorie: 2

0000000000

0CCC00000H

0CCC0H0000

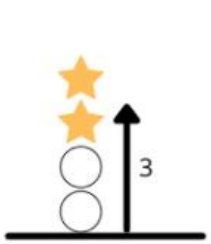
00000H0H0H

00000H0H0H

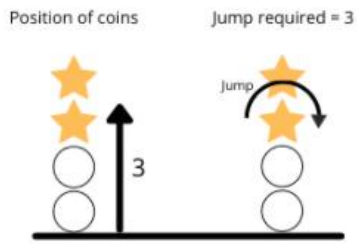
Column 0 is empty so Mario will only consume 1 calorie of energy. He was having 5 calories of energy, initially, and after crossing first column he will be left with 4 calories of energy.

We can see that we have coins on the screens at column 1, 2 and 3.

Column : 1,2,3



Coins



Coin Jump

The above two images describe collection of coins and energy spent in collecting them.

Column 1 has two coins at a height of 2 and 3 respectively. So, Mario will jump 3 units high and collect the highest coin. On his way down he will collect the coin at height 2. Total calories expended in collecting both coins in Column 1 is 3 * *JumpCalorie*. Since, he has also collected two coins, his calorie count will be replenished by 2 * *CoinCalorie*. Thus, Mario's calorie budget at the end of crossing column one is [Initial Calories - 1 - (3 * *JumpCalorie*) + (2 * *CoinCalorie*)].

Similar calculations will apply to columns 2 and 3.

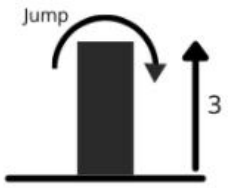
Substituting values of *JumpCalorie* and *CoinCalorie*, we get Mario's calorie budget equal to 7, after crossing column 3.

Column 4 is empty. So, 1 calorie is expended traversing it. Thus, we get Mario's calorie budget equal to 6, after crossing column 4.

Next, there are hurdles at Column 5 and 7.

There is a hurdle at column 5 of height 3.

Column : 5



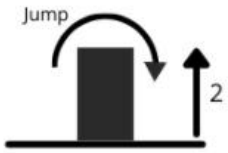
Hurdle

Calorie count after crossing column 5: 6 - 3 * *JumpCalorie**{1}*** = 3

Sixth column is empty, calorie count after crossing it will be: 3 - 1 = 2

There is a hurdle at column 7 of height 2.

Column : 7



Hurdle

Calorie count after crossing column 7: 2 - 2 * *JumpCalorie**{1}*** = 0

Now Mario's calorie budget is zero. He requires 1 calorie to cross column 8. Thus, he doesn't have enough energy for moving across column 8. Hence, print the column number successfully crossed i.e., 7 as output.

Constraints

$0 < I <= 200$

$0 < JumpCalorie < 10$

$0 < CoinsCalorie < 10$

$0 < M <= 10$

$10 <= N < 100$

Input

First line contains an integer I which denotes the initial calorie count.

Second line contains an integer ***JumpCalorie*** which denotes the value by which calorie is expended per unit of height.

Third line contains an integer ***CoinsCalorie*** which denotes the value by which calories are replenished per collected coin.

Next line contains two space separated integers M and N which depict the size of grid (screen).

Next M lines each contain a string of size N comprising of characters {0, C and H}.

Output

Output consists of two cases:

- If Mario is not able to cross the screen, print the last column number he has successfully crossed as output.
- If Mario is able to cross the screen successfully, then print the calorie budget he is having after crossing the screen as output.

Note: If Mario has crossed N - 1 column which is the last column, and he is left with a calorie budget of 0 calorie. Print 0 as output because he has crossed the screen successfully.

Time Limit (secs)

1

Examples

Example 1

Input

5

1

2

5 10

0000000000

0CCC00000H

0CCC0H0000

00000H0H0H

00000H0H0H

Output

7

Explanation:

Explained in problem description section.

Example 2

Input

11

1

2

5 10

0000000000

0000000000

0000000000

0000000000

0000000000

Output

1

Explanation:

Given in the input specification, we have

Initial Calorie count = 11

JumpCalorie: 1

CoinCalorie: 2

Screen size = 5 * 10

0000000000

0000000000

0000000000

0000000000

0000000000

But, as we see there are no coins, hurdles, or ring hurdles in the screen. Hence, the output will be initial calorie - (N * 1) = 1 i.e., the calorie remaining after crossing all the columns in the screen.