5 Easter Egg Hunt

You are participating in the annual Easter Egg Hunt! The game takes place inside a maze, where your goal is to collect as many hidden Easter eggs as possible and reach the exit — all within a limited number of moves.

You can move in four directions only — up, down, left, and right — but never diagonally or through walls. Additionally, once a position has been visited, it cannot be revisited—your algorithm must only explore new locations.

Your task is to find a path from the entrance to the exit that collects the **most eggs possible** while staying within the allowed number of moves. If no valid path to the exit exists, you should output N/A.

5.1 Input

The input consists of three parts:

- 1. The size of the maze as two integers (width, height) (e.g., 35,13, representing a 35×13 grid).
- 2. The number of **allowed moves** on the next line.
- 3. The maze itself, where:
 - Walls are denoted by +
 - Easter eggs are marked by **x**
 - The entrance is represented by I
 - The **exit** is represented by **E**

5.2 Output

The output should be a string representing the **best path** that collects the maximum number of eggs while successfully reaching the exit. The format consists of:

- A single integer indicating the number of eggs collected.
- A sequence of moves represented by:
 - U (Up)
 - D (Down)
 - L (Left)
 - R (Right)

If it is impossible to reach the exit, then output N/A

SAMPLE INPUT/OUTPUT ON NEXT PAGE

5.3 Sample Input/Output

Sample Input 1	Sample Output 1
7,5	1RDDRRRRR
8	
++++++	
I + x +	
+ + + +	
+ x E	
++++++	

Sample Input 2	Sample Output 2
5,5	N/A
10	
+++++	
I +	
+++++	
+ E	
+++++	

Explanation:

- In the first example, the best path collects 1 egg and successfully reaches the exit using the moves RDDRRRR. The egg at the top is unreachable within the 8-move limit, so the program takes a more optimal path to the exit.
- In the second example, the exit is completely blocked off, making it impossible to reach. The program detects this and correctly outputs N/A.