Write a program that, given a map with treasures and obstacles, computes the number of treasures that can be reached from a given initial position. The allowed movements are horizontal or vertical, but not diagonal. If needed, passing over the treasures is allowed.

**Input**

Input begins with the number of rows \( n > 0 \) and the number of columns \( m > 0 \) of the map. Follow \( n \) rows with \( m \) characters each. A dot indicates an empty position, an ‘\( X \)’ indicates an obstacle, and a ‘\( t \)’ indicates a treasure. Finally, two numbers \( r \) and \( c \) indicate the initial row and column (both of them starting at 1) where we must start looking for treasures. You can assume that \( r \) is between 1 and \( n \), that \( c \) is between 1 and \( m \), and that the initial position is always empty.

**Output**

Print the number of accessible treasures from the initial position.

**Sample input 1**

```
7 6
 ..t...
 ..XXX.
 ....
tX..X.
 .X..Xt
 .XX...
 ..t...
```

**Sample output 1**

```
4
```

**Sample input 2**

```
4 10
 ..t...X...
 ....X..t.
 XXXXX.X...
 .......X.t
```

**Sample output 2**

```
0
```

**Sample input 3**

```
5 7
 .......
 .XXXXXt
 .X...Xt
 .X.X.XX
 ...X.Xt
```

**Sample output 3**

```
2
```