Write a program that, given a map with treasures and obstacles, computes the distance from a given initial position to the nearest accessible treasure. The allowed movements are horizontal or vertical, but not diagonal.

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 minimum number of steps to reach a treasure starting from the initial position. If no treasure is accessible, tell so.

Sample input 1
```
7 6
..t...
..XXX.
......
tX..X.
.X..Xt
.XX...
..t...
5 3
```
Sample output 1
```
minimum distance: 4
```

Sample input 2
```
4 10
..t...X...
......X..t.
xxxxxxxx.x...
......X.t
4 1
```
Sample output 2
```
no treasure can be reached
```

Sample input 3
```
5 7
........
.XXXXXXt.
.X...Xt
.X.XXX
....X.Xt
5 5
```
Sample output 3
```
minimum distance: 19
```