Rajesh has found his old class photo which is an **MxN** grid where each cell consists of a face. Rajesh’s face is represented by ”\*” and his best friend is represented by “+” Rajesh wants to calculate the friendship index to his friend friendship index is defined as **N+M-G** Where **G** represents the Manhattan distance between the two of them.(Manhattan distance between two points is the absolute distance between the points coordinates) Print the friendship index between the two of them

**Input Format**

First Line consists of two space separated integers M and N the next M lines have N symbols (either . or \* or +)

**Constraints**

1<=M<=100 1<=N<=100

**Output Format**

Print a single integer which is the friendship index between Rajesh and his friend.

**Sample Input 0**

5 4

....

....

.+..

...\*

....

**Sample Output 0**

6

**Explanation 0**

The manhattan distance between + and \* in the given example is 3 hence the total value M+N-G is 6.

#include <cmath>

#include <cstdio>

#include <vector>

#include <iostream>

#include <algorithm>

using namespace std;

int main() {

int m,n,i;

cin>>m>>n;

int k;

int j1=m-1;

char a[20][20];

int x1=0,y1=0,x2=0,y2=0;

for(i=0;i<m;i++)

{ i=0;

for(k=0;k<n;k++)

{

cin>>a[i][j1];

//cout<<i<<" "<<j1<<endl;

i++;

}

i=0;

if(j1==0)

{

break;

}

else

{

j1--;

}

}

j1=m-1;

for(i=0;i<m;i++)

{ i=0;

for(k=0;k<n;k++)

{

if(a[i][j1]=='+')

{

x1=i;

y1=j1;

break;

}

else

{

i++;

}

}

i=0;

if(j1==0)

{

break;

}

else

{

j1--;

}

}

j1=m-1;

for(i=0;i<m;i++)

{ i=0;

for(k=0;k<n;k++)

{

if(a[i][j1]=='\*')

{

x2=i; y2=j1;

break;

}

else

{

i++;

}

}

i=0;

if(j1==0)

{

break;

}

else

{

j1--;

}

}

int c=abs(x1-x2)+abs(y1-y2);

cout<<m+n-c;

return 0;

}