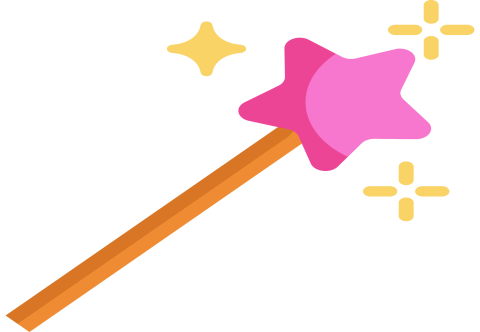
**PIYUSH AND MAGICAL PARK**

Piyush is lost in a magical park which contains N rows and M columns.In order to get out of park safely and return home, he needs atleast K amount of strength.Given a N by M pattern, your task is to find weather Piyush can ever escape the park.



Piyush enters the park with strength S. The park is filled with some obstacles denoted by ‘****.****’ , some magical beans denoted by ****‘\*’**** and some blockades denoted as ‘#’. If he encounters an obstacle (****.****) , strength decreases by 2. If he encounters a magic bean (****' \* '****) , his strength increases by 5. Piyush can only walk row wise, so he starts from left of a row and moves towards right and he does this for every row. However when he encounters a blockade (****#****) , he cannot go any further in his current row and simply jumps to the start of a new line without losing any strength. Piyush requires a strength of 1 for every step. His strength should always be greater than K while traversing or else Piyush will get lost. Assume that Piyush can shift immediately from last of one row to the start of next one without loss of any strength, help out Piyush to escape the park. His escape is successful if he is able to return home with atleast K strength.

**Input Format:**

First line of input contains four integers – N,M,K and S. Next N lines contains M space separated characters which can be '.', '\*' or '#'.

**Constraints:**

1 <= N,M,K,S <= 100

**Output Format**

Print "Yes" or "No" depending on whether Piyush can escape or not. If the answer is "Yes", also print the amount of strength he escaped with.

**Sample Input**

4 4 5 20

. . \* .

. # . .

\* \* . .

. # \* \*

**Sample Output**

Yes

13

Program-

#include<iostream>

using namespace std;

int main() {

int n,m,k,s;

cin>>n>>m>>k>>s;

char a[n][m];

for(int i=0;i<n;i++)

{

for(int j=0;j<m;j++)

{

cin>>a[i][j];

}

}

bool flag = true;

for(int i=0;i<n;i++){

for(int j=0;j<m;j++){

if(a[i][j]=='.'){

s=s-3;

if(j == m-1) s++; // if it is last element of the row we dont need to subtract energy for step

if(s <= k) flag = false; // checking if energy is greater than k or not

}

else if(a[i][j]=='\*'){

s=s+4;

if(j == m-1) s++; // if it is last element of the row we dont need to subtract energy for step

}

else if(a[i][j]=='#'){

break;

}

}

}

if(flag){

cout<<"Yes"<<endl;

cout<<s;

}

else{

cout<<"No"<<endl;

}

}