**Leroy's Game**

Leroy wants to play a game with his friend Lil'P but is bored of all the existing games. So he decides to create his own game. For that he takes a **N***x***M** board and colours some cells with the symbol **'\*'**. Being the math genius that he is, he creates a complex formula that he will use in his game. For that he needs to know how many coloured cells are neighbouring each cell.

Two cells are neighbours if they share a common edge or a common point, thus each cell can have upto 8 neighbours.

Since Leroy is busy playing snakes and ladders with Lil'P, he has assigned you the task of calculating the coloured neighbours of each of the cells of **T** boards.

**Input Format**

First line of the input format **T** number of boards *1 <= T <= 100*

Each board contains:

* First line of the board contains it dimensions **N***x***M** as space separeated integers *1 <= N, M <= 100*
* Next *N* lines contains *M* characters each with either **'.'** or **'\*'**

**Output Format**

Print each board on **N** lines with **M** characters per line, and replace every *character* with the appropriate digit indicating the number of neighbouring cells that are coloured.

**Sample Input**

1  
3 2  
..  
.\*  
..

**Sample Output**

11  
10  
11

**Explanation**

Cells (1,1), (3,1) are neighbouring 1 **'\*'** sharing a point.

Cells (1,2), (2,1), (3,2) are neighbouring 1 **'\*'** sharing an edge.

Cell (2,2) doesn't contain any coloured cell neighbour

Solution :

import java.io.\*;

import java.util.\*;

public class Solution {

public static void main(String[] args) {

Scanner sc = new Scanner(System.in);

int t=sc.nextInt();

for(int x=0;x<t;x++){

int n=sc.nextInt();

int m=sc.nextInt();

String m\_accept[]=new String[n];

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

m\_accept[i]=sc.next();

}

char m\_accept\_split[][]=new char[n][m];

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

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

m\_accept\_split[i][j]=m\_accept[i].charAt(j);

}

}

int b[][]=new int[n][m];

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

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

b[i][j]=0;

}

}

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

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

if(m\_accept\_split[i][j]=='\*')

{

if((j-1)>=0)

{ b[i][j-1]+=1; }

if((j+1)<m)

{ b[i][j+1]+=1; }

if((i-1)>=0)

{ b[i-1][j]+=1; }

if((i+1)<n)

{ b[i+1][j]+=1; }

if((j-1)>=0 && (i-1)>=0)

{ b[i-1][j-1]+=1; }

if((j+1)<m && (i+1)<n)

{ b[i+1][j+1]+=1; }

if((j-1)>=0 && (i+1)<n)

{ b[i+1][j-1]+=1; }

if((j+1)<m && (i-1)>=0)

{ b[i-1][j+1]+=1; }

}

}

}

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

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

System.out.print(b[i][j]);

}

System.out.println();

}

}

}

}