Prva

Little Ivica solves crossword puzzles every day. In case you haven’t seen one, a crossword puzzle starts on a grid ofR×CR×C squares, each of which is either empty or blocked. The player’s task is to write words in consecutive empty squares vertically (top down) or horizontally (left to right).

Ivica’s sister has a strange habit of looking at crosswords Ivica has finished solving, and finding the **lexicographically smallest word** in it. She only considers words at least 22 characters long.

Write a program that, given a crossword puzzle, finds that word.

**Input**

The first line contains two integers RR and CC (2≤R,C≤202≤R,C≤20), the number of rows and columns in the crosswords.

Each of the following RR lines contains a string of CC characters. Each of those characters is either a lowercase letter of the English alphabet, or the character ‘#’ representing a blocked square.

The input will be such that a solution will always exist.

**Output**

Output the lexicographically smallest word in the crossword.

|  |  |
| --- | --- |
| **Sample Input 1** | **Sample Output 1** |
| 4 4  luka  o#a#  kula  i#a# | kala |

|  |  |
| --- | --- |
| **Sample Input 2** | **Sample Output 2** |
| 4 4  luka  o#a#  kula  i#as | as |

|  |  |
| --- | --- |
| **Sample Input 3** | **Sample Output 3** |
| 4 5  adaca  da##b  abb#b  abbac | abb |

<https://open.kattis.com/problems/prva>

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

namespace ConsoleApplication1

{

class Program

{

static void Main(string[] args)

{

//string[] s = {

// "luka",

// "o#a#",

// "kula",

// "i#a#"

//};

//string[] s = {

// "aaa",

// "aaa",

// "aa#",

// "aaa"

//};

//string[] s = {

//"adaca",

//"da##b",

//"abb#b",

//"abbac"

// } ;

string[] rc = Console.ReadLine().Trim().Split(' ');

int r = int.Parse(rc[0]);

int c = int.Parse(rc[1]);

string[] s = new string[r];

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

{

s[i] = Console.ReadLine().Trim();

}

List<string> lista = new List<string>();

for (int i = 0; i < s.Length; i++)

{

string concat = "";

for (int j = 0; j < s[i].Length; j++)

{

if (s[i][j] == '#')

{

if (concat.Length >= 2)

{

lista.Add(concat);

}

concat = "";

continue;

}

concat += s[i][j];

}

if (concat.Length >=2) lista.Add(concat);

}

for (int j = 0; j < s[0].Length; j++)

{

string concat = "";

for (int i = 0; i < s.Length; i++)

{

if (s[i][j] == '#')

{

if (concat.Length >= 2)

{

lista.Add(concat);

}

concat = "";

continue;

}

concat += s[i][j];

}

if (concat.Length >=2) lista.Add(concat);

}

//lista.Sort();

string ans = lista[0];

for (int i = 1; i < lista.Count; i++)

{

if (lista[i].CompareTo(ans) < 0)

{

ans = lista[i];

}

}

Console.WriteLine(ans);

//foreach (string item in lista)

//{

// Console.WriteLine(item);

//}

Console.ReadLine();

}

}

}