Encryption

by HackerRank

Problem Submissions Leaderboard Discussions Editorial

An English text needs to be encrypted using the following encryption scheme.
First, the spaces are removed from the text. Let L be the length of this text.
Then, characters are written into a grid, whose rows and columns have the following constraints:

- $\lfloor\sqrt{L}\rfloor \leq rows \leq column \leq \lceil\sqrt{L}\rceil$, where $\lfloor x\rfloor$ is floor function and $\lceil x\rceil$ is ceil function

For example, the sentence `if man was meant to stay on the ground god would have given us roots` after removing spaces is 54 characters long, so it is written in the form of a grid with 7 rows and 8 columns.

```
ifmanwas
meanttos
tayonthe
groundgo
dwouldha
vegivenu
sroots
```

- Ensure that $rows \times columns \geq L$
- If multiple grids satisfy the above conditions, choose the one with the minimum area, i.e. $rows \times columns$.

The encoded message is obtained by displaying the characters in a column, inserting a space, and then displaying the next column and inserting a space, and so on. For example, the encoded message for the above rectangle is:

`imtgdvs fearwer mayoogo anouuio ntnnlvt wttddes aohghn sseoau`

You will be given a message in English with no spaces between the words. The maximum message length can be 81 characters. Print the encoded message.

Here are some more examples:

Sample Input:

```
haveaniceday
```

Sample Output:

```
hae and via ecy
```

Sample Input:

```
feedthedog
```

Sample Output:

```
fto ehg ee dd
```

Sample Input:

```
chillout
```

Sample Output:

```
clu hlt io
```

Submissions: 22998
Max Score: 30
Difficulty: Medium

Rate This Challenge:

More

Current Buffer (saved locally, editable) C#

```
using System;
using System.Collections.Generic;
using System.IO;
using System.Linq;
class Solution {

    static void Main(String[] args) {

        string s =  Console.ReadLine(); // "if man was meant to stay on the ground god would have given us roots";

            s = s.Replace(" ", "");
            //Console.WriteLine(s);

            int rows = (int)Math.Floor(Math.Sqrt(s.Length));
            int cols = (int)Math.Ceiling(Math.Sqrt(s.Length));
            //Console.WriteLine(rows + " " + cols);

            if (rows * cols < s.Length)
            {
                rows = cols;
            }

            char[,] matriz = new char[rows, cols];
            int index = 0;

            for (int i = 0; i < rows ; i++)
            {
                for (int j = 0; j < cols ; j++)
                {
                    if (index < s.Length)
                    {
                        matriz[i, j] = s[index];
                    }
                    if (index < s.Length) index++;
                }
            }

            //for (int i = 0; i < rows; i++)
            //{
            //    for (int j = 0; j < cols; j++)
            //    {
            //        Console.Write(matriz[i, j] + "");
            //    }
            //    Console.WriteLine();
            //}

            string result = "";
            for (int j = 0; j < cols; j++)
            {
                string col = "";
                for (int i = 0; i < rows; i++)
                {
                    //Console.Write(matriz[i, j]);
                    if (matriz[i, j] != '\0')
                    {
                        col += matriz[i, j];
                    }
                }
                result += col + " ";
                //Console.Write(col + " ");
            }
            //result = result.Replace("  ", " ");
```

```
62              //result = result.Replace(    ,    );
63              Console.WriteLine(result);
64
65
66
67      }
68  }
69
```

Line: 51 Col: 17

Upload Code as File Test against custom input Run Code Submit Code

Congrats, you solved this challenge!

- ✔ Test Case #0
- ✔ Test Case #1
- ✔ Test Case #2
- ✔ Test Case #3
- ✔ Test Case #4
- ✔ Test Case #5
- ✔ Test Case #6
- ✔ Test Case #7
- ✔ Test Case #8
- ✔ Test Case #9
- ✔ Test Case #10
- ✔ Test Case #11
- ✔ Test Case #12

Next Challenge

Copyright © 2016 HackerRank. All Rights Reserved