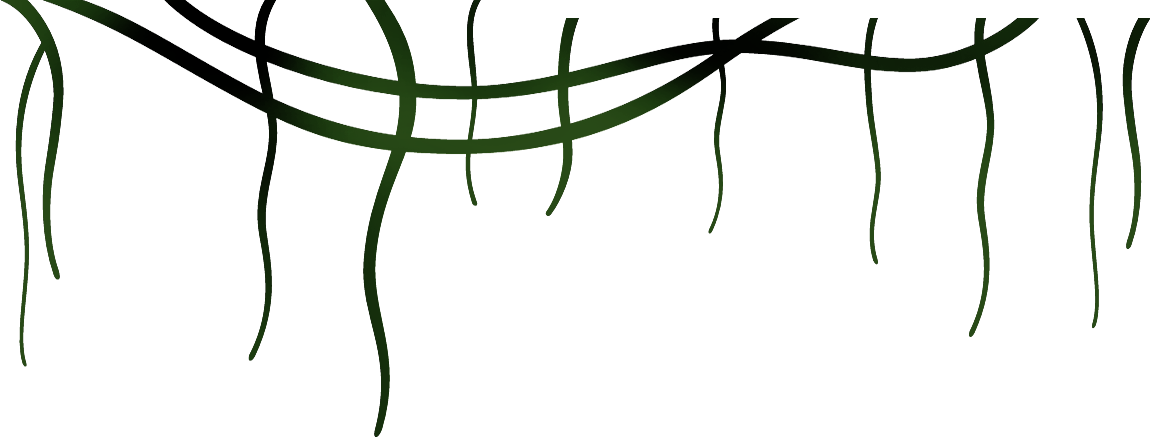
# C:\Users\Ivo\AppData\Local\Microsoft\Windows\INetCache\Content.Word\vine-rope.pngC:\Users\Ivo\AppData\Local\Microsoft\Windows\INetCache\Content.Word\vine-rope.pngC:\Users\Ivo\AppData\Local\Microsoft\Windows\INetCache\Content.Word\vine-rope.pngC:\Users\Ivo\AppData\Local\Microsoft\Windows\INetCache\Content.Word\vine-rope.pngProblem 3 – NMS



Furion has created the NMS – Natural Messaging Service, for all the flowers. They communicate through special messages which only they understand. You want to know what the flowers are saying to each other, that’s why you’ve decided to create a software program which translates the messages.

You will be given several input lines of random, **upper case** and **lower case, English alphabet letters**. You need to find **all words** in that message. A word in the Natural language is an **increasing sequence of letters**.

Тhe message **“abc”** is a **single** word, because **“b”>“a”** and **“c”>“b”**, therefore it **IS** an increasing sequence and it is counted as a word. The message **“abca”** consists of **2** words – **“abc”** and **“a”** because **“a”<“c”** and it **breaks** the increasing sequence and **begins a new one.   
Equal letters** do **NOT** break the subsequence. The **comparison** is **case-insensitive**.

The input **ends** when you receive the command **“---NMS SEND---“**. After it you will receive a **specified delimiter**. You need to **break the whole message** into **words** and print them as a text, each separated with **the given delimiter**. The delimiter can be a line of **random ASCII characters** with **random length**.

NOTE: The **comparison** is **case-insensitive**, as specified above, therefore **“B”>“a”**, and **“A”=”a”**.

### Input

* You will be receiving lines of input containing random English alphabet letters, until you receive the line –   
  **“---NMS SEND---“**.
* After you receive the ending command, you will receive the delimiter on the next line, as **the last line of input**.

### Output

* As output you need to print all the words you’ve found, **separated by the given delimiter**.

### Constraints

* All of the input lines, except theinput-terminating one and the delimiter, will consist only of uppercase and lowercase English alphabet letters.
* The maximum lines of input you can receive is 1000.
* The delimiter will be a string, which can consist of any ASCII character.
* Allowed time/memory: 100ms/16MB



### C:\Users\Ivo\AppData\Local\Microsoft\Windows\INetCache\Content.Word\vine-rope.pngC:\Users\Ivo\AppData\Local\Microsoft\Windows\INetCache\Content.Word\vine-rope.pngExamples

|  |  |
| --- | --- |
| **Input** | **Output** |
| **Foxtrot**  **Uniform**  **Charlie Kilo**  **---NMS SEND---**  **(space)** | **Fox t r otU n i for m Ch ar l i eK ilo** |

|  |  |
| --- | --- |
| **Input** | **Output** |
| **Abcdefghijklmnopqrstuvwxyz**  **ABCDEFGHIJKLMNOPQRSTUVWXYZ**  **---NMS SEND---**  **---NMS SEND---** | **abcdefghijklmnopqrstuvwxyz---NMS SEND---ABCDEFGHIJKLMNOPQRSTUVWXYZ** |

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace Nms3

{

class Program

{

static void Main()

{

StringBuilder command = new StringBuilder();

string sb = "";

string input = Console.ReadLine();

while (input != "---NMS SEND---")

{

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

{

char current = input[i];

if (char.IsLetterOrDigit(current))

{

sb += (current);

command.Append(current);

}

}

input = Console.ReadLine();

}

string delimeter = Console.ReadLine();

StringBuilder result = new StringBuilder();

result.Append(sb[0]);

sb = sb.ToLower();

for (int i = 0; i < sb.Length - 1; i++)

{

char current = sb[i];

char currentNext = sb[i + 1];

if (current <= currentNext)

{

result.Append(command[i+1]);

}

else

{

result.Append(delimeter);

result.Append(command[i+1]);

}

}

Console.WriteLine(result.ToString());

}

}

}