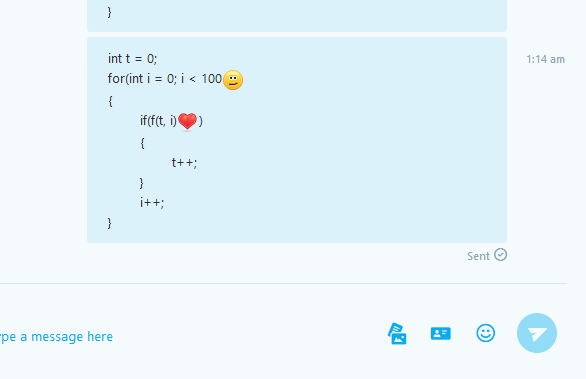
Sometimes, to make an on-line conversation more expressive, we use some smileys (or emoticons) in our messages. What actually happens is this: we use some notations that represent emoticons, and the chat system converts them into images one a message is sent.

To replace a keyword with an emoticon, the system needs to correctly detect it first. Your mission is to write a function that will detect emoticons in a message. Given a message and emoticons, the function should detect all emotions in the message and enclose each of them in a pair of brackets ([ and ]).

Remember this: an emotion should be replaced only if it is indeed an emoticon, i.e. if it is a word that is not a part of some other word. Thus, the function shouldn't detect non-emoticons and ruin messages like in the image below:  


Good luck, :P :D ;)

**Example**

* For message = "I love you <3" and emoticons = ["<3", ":)", "^\_^"],  
  the output should be  
  Emojticon(message) = "I love you [<3]".

<3 in the given message is, indeed, an emoticon.

* For message = "I love you because 1<3" and  
  emoticons = ["<3", ":)", "^\_^"], the output should be  
  Emojticon(message) = "I love you because 1<3".

Here <3 is not an emoticon, so it shouldn't be replaced.

**Input/Output**

* **[time limit] 3000ms (cs)**
* **[input] string message**

A messages in the conversation.

*Constraints*  
1 ≤ message.length ≤ 100.

* **[input] array.string emoticons**

A set of emoticons that should be replaced. It is guaranteed that the emoticons are unique, and that no emoticon is a substring of another emoticon.

*Constraints:*  
1 ≤ emoticons.length ≤ 10,  
1 ≤ emoticons[i].length ≤ 5.

* **[output] string**

The message with emoticons enclosed in brackets.

<https://codefights.com/challenge/zwoGe7JRnS43tSbvm?utm_source=featuredChallenge&utm_medium=email&utm_campaign=email_notification>

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

namespace ConsoleApplication1

{

class Program

{

//-------mi solucion-------------

static string Emojticon(string message, string[] emoticons)

{

message = " " + message + " ";

foreach (string emot in emoticons)

{

string em = " " + emot + " ";

int indice = message.IndexOf(em);

while (indice != -1)

{

message = message.Remove(indice, em.Length);

message = message.Insert(indice, " [" + emot + "] ");

indice = message.IndexOf(em, indice + ("[" + emot + "] ").Length);

}

}

return message.Substring(1, message.Length - 2) ;

}

static void Main(string[] args)

{

//string message = "I love you <3";

//string[] emoticons = { "<3", ":)", "^\_^" };

//string message= "I love you because 1<3";

//string[] emoticons = { "<3", ":)", "^\_^" };

//string message= "<3 3:) :) :( =)) ;)";

//string[] emoticons = { "<3", ":3", "3:)", ";)" };

//string message = ":\* .. kk 53 d2 li 3. :. ;) :) :):) :( <3 :\* .\* ... .. :? ;[ :] 8] 8} {4} {3} {3";

//string[] emoticons = { ":)", ":(", "<3", ":\*", "..", ":?", ";[", ":]", "8}", "{3" };

////expected = "[:\*] [..] kk 53 d2 li 3. :. ;) [:)] :):) [:(] [<3] [:\*] .\* ... [..] [:?] [;[] [:]] 8] [8}] {4} {3} [{3]"

//Console.WriteLine(Emojticon(message, emoticons));

string message = ":D :D" ;

string[] emoticons = {":D"}; // => "[:D] [:D]"

Console.WriteLine(Emojticon(message, emoticons));

//string em = "{3";

//int indice = message.IndexOf(em);

//Console.WriteLine( message[indice + em.Length]);

//Console.WriteLine(Emojticon(message, emoticons).Length);

//Console.WriteLine(message.Length);

Console.ReadLine();

}

}

}

//----solucion por m\_reyes------

string Emojticon(string message, string[] emoticons)

{

string[] sa = message.Split(' ');

foreach (string s in emoticons)

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

if (sa[i] == s)

sa[i] = "[" + s + "]";

return string.Join(" ", sa);

}