Write a function that reverses characters in (possibly nested) parentheses in the input string.

Input strings will always be well-formed with matching ()s.

Example

* For inputString = "(bar)", the output should be  
  reverseInParentheses(inputString) = "rab";
* For inputString = "foo(bar)baz", the output should be  
  reverseInParentheses(inputString) = "foorabbaz";
* For inputString = "foo(bar)baz(blim)", the output should be  
  reverseInParentheses(inputString) = "foorabbazmilb";
* For inputString = "foo(bar(baz))blim", the output should be  
  reverseInParentheses(inputString) = "foobazrabblim".  
  Because "foo(bar(baz))blim"becomes "foo(barzab)blim" and then "foobazrabblim".

Input/Output

* **[execution time limit] 3 seconds (cs)**
* **[input] string inputString**

A string consisting of lowercase English letters and the characters ( and ). It is guaranteed that all parentheses in inputString form a [regular bracket sequence](keyword://regular-bracket-sequence).

*Guaranteed constraints:*  
0 ≤ inputString.length ≤ 50.

* **[output] string**
  + Return inputString, with all the characters that were in parentheses reversed.

**[C#] Syntax Tips**

// Prints help message to the console

// Returns a string

**string** **helloWorld**(**string** name) {

Console.Write("This prints to the console when you Run Tests");

**return** "Hello, " + name;

}

<https://app.codesignal.com/arcade/intro/level-3/9DgaPsE2a7M6M2Hu6/description>

using System;

using System.Collections.Generic;

using System.IO;

using System.Linq;

class Solution

{

static void ReverseSubArray(char[] arr, int ini, int fin)

{

int i = ini;

int j = fin;

while (i < j)

{

char temp = arr[i];

arr[i] = arr[j];

arr[j] = temp;

i++;

j--;

}

}

static string reverseInParentheses(string inputString)

{

char[] ch = inputString.ToCharArray();

Stack<int> stack = new Stack<int>();

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

{

if (ch[i] == '(')

{

stack.Push(i + 1);

}

else if (ch[i] == ')')

{

int ini = stack.Pop();

//string subs = inputString.Substring(ini, ini + i);

ReverseSubArray(ch, ini, i - 1);

}

}

string ans = "";

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

{

if(ch[i] != '(' && ch[i] != ')')

{

ans += ch[i];

}

}

return ans;

}

static void Main(String[] args)

{

//char[] cad = "abcdefghijkl".ToArray();

//ReverseSubArray(cad, 3, 8);

//Console.WriteLine(new string(cad));

string inputString = "foo(bar(baz))blim";

//"foobazrabblim"

string res = reverseInParentheses(inputString);

Console.WriteLine(res);

Console.ReadLine();

}

}