Author

[madhuram\_j](https://codefights.com/profile/madhuram_j)

https://codefights.com/img/coins_new.png3000

You're given a sentence, where each word has different length. Swap the longest word with the shortest one, the 2nd shortest word with the 2nd longest one, and so on.

Note that the resulting sentence should be correct, i.e. it should be capitalized and all words but the first one should contain only lowercase letters (except the word"I").

**Examples:**

* replaceWords("I am the best") = "Best the am I"
* replaceWords("I am better than him") = "Better than I am him"
* **[input] string sentence**
  + A string of English letters and whitespace characters, 0 < sentence.length < 200.
* **[output] string**
  + Correct sentence with words replaced as described above.

<https://codefights.com/challenge/TsBHzguYB32YKujuG>

static string replaceWords(string sentence)

{

string[] palabras = sentence.Split(' ');

//hago un array de lengths y lo cargo

//con la longitud de cada palabra

int[] lengts = new int[palabras.Length];

for (int k = 0; k < palabras.Length; k++)

{

lengts[k] = palabras[k].Length;

}

//hago un array de copia para que pueda ordenarlo y en el de lengths

//se conserve la posicion original

int[] copiaLengts = new int[lengts.Length];

Array.Copy(lengts, copiaLengts, lengts.Length);

//ordeno el array de copia lengths y ordenar sirve para que

//con un puntero i y j intercambie el primero con el ultimo

//el 2do con el ultimo-1, el 3ro con el ultimo-2,

//porque asi intercambio el de mayor longitud con el de menor,

//el que le sigue en longitud mayor con el que le sigue en longitud menor

Array.Sort(copiaLengts);

int i = 0, j = copiaLengts.Length - 1;

while (i < j)

{

//intercambio el elemento de palabras,

//que esta como elemento en copia lengths, pero que saco el

//indice de ese elemento en lengts, porque lengths es un arrya paralelo

//a palabras donde se guarda en el la longitud de cada palabra en palabras

string temp = palabras[Array.IndexOf(lengts, copiaLengts[i])];

palabras[Array.IndexOf(lengts, copiaLengts[i])] = palabras[Array.IndexOf(lengts, copiaLengts[j])];

palabras[Array.IndexOf(lengts, copiaLengts[j])] = temp;

i++;

j--;

}

string answer = "";

for (i = 0; i < palabras.Length; i++)

{

if (palabras[i] != "I") //en los test cases pide que se conserve la I mayuscula

{

palabras[i] = palabras[i].ToLower();

}

}

//y pide que la nueva primera letra de toda la sentencia sea mayuscula

palabras[0] = char.ToUpper(palabras[0][0]) + palabras[0].Substring(1,palabras[0].Length-1);

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

{

answer += palabras[i] + " ";

}

answer += palabras[i];

return answer;

}