Find all occurrences of the substring in the given string and replace them with another given string...  
just for fun :)

**Example:**  
findAndReplace("I love Codefights", "I", "We") = "We love Codefights"

* **[input] string originalString**
  + The original string.
* **[input] string stringToFind**
  + A string to find in the originalString.
* **[input] string stringToReplace**
  + A string to replace with.
* **[output] string**
  + The resulting string.

<https://codefights.com/challenge/fedrX82u6EMwxvhmJ>

--ACEPTADO--

std::string findAndReplace(std::string originalString, std::string stringToFind, std::string stringToReplace) {

std::vector<int> indices; // = new List<int>();

for (int i = 0; i < originalString.length() - stringToFind.length() +1; i++)

{

if (originalString[i] == stringToFind[0])

{

if (originalString.substr(i, stringToFind.length()) == stringToFind)

{

indices.push\_back(i);

}

}

}

std::string res = "";

int j = 0, k=0;

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

while (k < originalString.length())

{

if (j < indices.size() && k == indices[j])

{

res += stringToReplace;

j++;

k += stringToFind.length()-1;

}

else

{

res += originalString[k];

}

k++;

}

return res ;

}

--EN C#--

static string findAndReplace(string originalString, string stringToFind, string stringToReplace)

{

List<int> indices = new List<int>();

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

{

if (originalString[i] == stringToFind[0])

{

if (originalString.Substring(i, stringToFind.Length) == stringToFind)

{

indices.Add(i);

}

}

}

string res = "";

int j = 0, k = 0;

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

while (k < originalString.Length)

{

if (j < indices.Count && k == indices[j])

{

res += stringToReplace;

j++;

k += stringToFind.Length - 1;

}

else

{

res += originalString[k];

}

k++;

}

return res;

}