

gibberish

mhb93

October 2020

Function 2: Remove Character The program will need to scan the first given string for the element given in the second string, then remove said second element from the first string. I'm thinking I'll need a list to append all the "good string indexes" to, then I can concatenate the elements in that list and return the concatenation. Post-write In a stroke of genius (and because it was shorter than just concatenating the elements in a string together), I just created a string that would add each "correct" component of the string.

Function 3: Reverse This will need a scanning pattern that works in reverse. I will set the variable i to the last index of the list and work backwards until it is at the first. As i goes, it will add each letter to a single new string. Post-write Works like a charm!

Function 4: Palindrome Detection This code will work a lot like the reverse() function. Once the string is reversed, I will compare it to the given string and have the function return true if they are the same. Post-write That took a lot more thought than I thought it would. I had to remove undesirable, non-essential characters (like spaces and punctuation), which made use of my remove_character() function. In retrospect, I should have used the reverse() function. In fact, I'll go do it now. Post-write I did it.

Function 5: String to ASCII Codes Since there is a function that returns the value that I need, I just have to create a list to house all of them. And then return the list. Post-write Works like a charm!

Function 6: Max I will need to scan through the list and find the biggest number. I can have this number be a string or have its own list, as long as it gets returned at the end. I will need to be able to dispose of a number that was the biggest but isn't anymore; with a list I can .pop(), and with a string I can just re-value it. Post-write In creating my "while" statement (and the "if" statement I had to add to it), I had to define a value (append_me) before I had an actual value to assign it to; so, I set it equal to a very small number (or a big negative number).

Function 2: Remove Character The program will need to scan the first given string for the element given in the second string, then remove said second element from the first string. I'm thinking I'll need a list to append all the "good string indexes" to, then I can concatenate the elements in that list and return the concatenation. Post-write In a stroke of genius (and because it was shorter than just concatenating the elements in a string together), I just created a string that would add each "correct" component of the string.

Function 3: Reverse This will need a scanning pattern that works in reverse. I will set the variable i to the last index of the list and work backwards until it is at the first. As i goes, it will add each letter to a single new string. Post-write Works like a charm!

Function 4: Palindrome Detection This code will work a lot like the reverse() function. Once the string is reversed, I will compare it to the given string and have the function return true if they are the same. Post-write That took a lot more thought than I thought it would. I had to remove undesirable, non-essential characters (like spaces and punctuation), which made use of my remove_character() function. In retrospect, I should have used the reverse() function. In fact, I'll go do it now. Post-write I did it.

Function 5: String to ASCII Codes Since there is a function that returns the value that I need, I just have to create a list to house all of them. And then return the list. Post-write Works like a charm!

Function 6: Max I will need to scan through the list and find the biggest number. I can have this number be a string or have its own list, as long as it gets returned at the end. I will need to be able to dispose of a number that was the biggest but isn't anymore; with a list I can .pop(), and with a string I can just re-value it. Post-write In creating my "while" statement (and the "if" statement I had to add to it), I had to define a value (append_me) before I had an actual value to assign it to; so, I set it equal to a very small number (or a big negative number).