



Tutorial 8 : Strings

Exercise 1:

Write a C function that receives as parameters a string S and a character c and returns the number of occurrences of c in S.

Exercise 2:

Write a program that reads a string that can contain spaces and prints each word of the string, with the space as the separator.

Example: the user enter: manipulating strings in C

The program displays :

word 1 : manipulating

word 2 : strings

word 3 : in

word 4 : C

Exercise 3:

Write a C program that reads a string S and checks if it is a palindrome or not. A palindrome is a string that reads the same backward as forward. Example: radar, madam, 1234321.

If the string S is not a palindrome, the program must find the longest substring of S that is a palindrome.

Exercise 4:

Write a C program to capitalize the beginnings of words in a sentence.

Example : this is a string of characters

output: This Is A String Of Characters

Exercise 5:

Write a C program to find the first longest substring without repetition in a string entered by the user.

Example : abcddefghhijk

output: defgh

Exercise 6:

Write a C function to replace any sequence of the same character in a string by a single character.

Example:

For the string S = "AABCCC DD EEEFFG"

The function must return S = "ABC D EFG"

Exercise 7:

Write a C function to read a string S and insert it into a sorted array T of N strings in ascending order (alphabetical order) so that the array T remains sorted. N is given as a parameter. Note that the strings themselves do not move.

The memory required to implement each string must be allocated as it is read with the function malloc.