

06. Strings

6.01. Write an implementation of:

- strlen;
- strcmp;
- strcat;
- strcpy.

6.02. Write a program that searches for start indexes of all occurrences of a given substring in a given string.

Example input	Expected output
canary a	1 3
Jake and Jill and their friend have androids and	5 14 36

6.03. Write the following program: You are given two strings separated by a new line. Each string will consist of lower case Latin characters ('a'-'z').

- In the first line print two space-separated integers, representing the length of a and b respectively;
- In the second line print the string produced by concatenating a and b;
- In the third line print two strings separated by a space, a' and b'. a' and b' are the same as a and b, respectively, except that their first characters are swapped;

Example input	Expected output
abcd ef	4 2 abcdef ebcd af

6.04. Write a function that checks whether a given word is a palindrome.

Example input	Expected output
civic	1
palindrome	0
kapak	1

6.05. Write a function with signature `void revert(char* text)`, which converts the given string as a parameter, replacing all lowercase Latin alphabets with uppercase and reverse. Do not use the built-in string library.

6.06. Write a program which reads a line containing integer numbers, separated by spaces, and prints their sum. In addition to the numbers, each line will contain one or more words (sequences of English letters) – print those words on a separate line, separated by spaces, after the sum, in the order they were in the input.

Example input	Expected output
1 2 3 invalid 4	10 invalid
foo 2 bar baz -1 4	5 foo bar baz
0 HELLO 13 -5 ten 10 14 Noise	32 HELLO ten Noise

6.07. Write a function which checks whether a string is a substring in another string. If true, the functions must return the index to the first position the substring is located in the string, else return -1.

6.08. Write a program which changes each word in a text to start with a capital letter (don't bother with the exact title-case rules about not capitalizing things like in, from, etc.). Assume the first letter of a word is an English alphabetical symbol preceded by a non-alphabetical symbol (so in "we will--rock you", "we", "will", "rock" and "you" are all considered words which need to be capitalized).

Example input	Expected output
On the south Carpathian mountains, a tree is swinging	On The South Carpathian Mountains, A Tree Is Swinging
Write a program which changes each word in	Write A Program Which Changes Each Word In

6.09. Write a program which is given a line of text, another line with a "find" string and another line with a "replace" string. Any part of text which matches the "find" string should be replaced with the "replace" string. Print the resulting text on the console.

Example input	Expected output
I am the night. Dark Night! No, not the knight night day	I am the day. Dark Night! No, not the kday

6.10. Write a function which counts words in `char* text`. A "word" is defined to be a single or a sequence of characters different from ' ', '\t', '\n'. If the array of characters is empty, the function must return -1.

6.11. Write a function which takes a string and prints out a char histogram of that string. Each string will consist of lower case Latin characters ('a'-'z').

Example input	Expected output
aaaabb	a - 4 b - 2
randomness	r - 1 a - 1 n - 2 d - 1 o - 1 m - 1 e - 1 s - 2

NOTE: The order of the output is irrelevant. The only importance is the histogram accuracy.

github.com/andy489