



## TD7 Character strings

### Ex 7.1 Upper/Lower case

Write a program that converts a character string to upper or lower case, depending on the user's preference, and displays the result.

### Ex 7.2 Counting

1. *Counting Vowels*: Write a program that counts the number of vowels in a given character string.
2. *Counting occurrences of a character*: Write a program that counts the number of occurrences of a specific character in a string given by the user, and then deletes all these occurrences from the string.

You have to implement two functions:

- A function to count the occurrences of a character in a string.
- A function to remove all occurrences of a character from the string.

*Example* : String : "Hello world !"

Character : 'o'

The number of occurrences of the character 'o' is 2.

After deleting all occurrences of the character 'o' your string becomes : "Hell wrld !"

3. *Counting words*: Write a program that counts the number of words in a given sentence. You can assume that words are separated by spaces.

### Ex 7.3 Palindrome

Write a program that checks whether a character string given by the user is a palindrome.

A palindrome is a sequence of characters that can be read in the same way from left to right and right to left. *For example* : radar, level, civic, noon, etc.

The program must include two functions:

1. A function **ReverseString** that takes a character string as input and returns the same string with the characters in reverse order.
2. A function **IsPalindrome** that tests whether a character string is a palindrome, returning 1 if it is and 0 otherwise.

Your program should display an appropriate message indicating whether the string is a palindrome or not.

### Ex7.4 Cryptographie

1. Write a **Caesar** function that allows you to encode a text, entered by the user, using a shift cipher. Shift cipher (also known as Caesar's cipher) involves shifting each letter of the initial text by a specified number of positions in the alphabet.

The encoded text should be displayed on the screen.

- *Notes* : \* The alphabet is circular, i.e. after 'Z' we return to 'A'.  
\* The other characters in the message are not modified.
- *For example*, if the user enters the text "Hello World!" and specifies an offset of 3, the encoded text should be "Khoor Zruog!".

2. Another cryptographic technique involves random substitution. To do this, we use a "key-alphabet", in which the letters follow one another in a disorderly way.

*For example :*

The key-alphabet "HYLUJVPREAKBCDOISQZNXGFTM" will be used to encode the message.

According to our example, A will become H, B will become Y, C will become L, etc.

Write a function **Code** that performs this encryption (the key-alphabet will be passed as a parameter).

3. Write a main function to test the above functions.

### Ex 7.5 Email address validation

Write a program that takes an email address as input and checks if it is valid according to the following criteria:

➤ Email address:

- Must contain exactly one character '@'.
- Must contain at least one character '.' after the '@'.
- Must contain at least one character before and after the '@'.
- must not begin with '@'.
- must not end with '.'.
- Must be at least 6 characters in length.

If the email address meets all these criteria, the program should display "The email address is valid", otherwise, it should display "The email address is invalid".

### Ex 7.6 Word Search (Optional)

1. *Search and replace words:* Write a program that takes a sentence and two words as input, then replaces all occurrences of the first word with the second word in the sentence, and displays the resulting sentence.
2. *Deleting a word from a sentence:* Write a program that deletes a user-given word from a given sentence, and displays the resulting sentence.
3. *Word frequency calculation:* Write a program that takes a sentence as input and calculates the frequency of each word in the sentence, then displays the words with their frequency.