
CSCE 1101 Spring 2023: Fundamentals of Computing II Assignment #1

Prof. Amr Goneid

Date: Mon Feb 6, Due: Mon Feb 20, 2023

This assignment will serve to revise the C++ programming material given in CSCE 1001

Assignment instructions:

- 1- Each problem is answered in a CPP file named problem-x.cpp
- 2- Submit all CPP files on Bb. Don't submit a Zip file.
- 3- Write your name and ID in each CPP file.
- 4- Your comments inside the code will be weighted 10% of the assignment.
- 5- Keep your eyes on the due date, as the late policy will be applied. [Check policy](#)
- 6- Corrupted programs will be ZERO Graded; submit running programs.

Problem 1: (5 points)

Write a function **uppercase** to receive an alphabetic character and return it in uppercase. For example, if the received character is 'a' it returns 'A', and if it is 'A' it returns 'A'. (The ASCII code for 'A' is 65, for 'B' it is 66, etc, while for 'a' it is 97 and for 'b' it is 98, etc).

Hint: Use ASCII code computation to get the full mark.

Problem 2: (5 points)

Write a function that receives the number of seconds up to 18,000 (5 hours) and returns the equivalent hours, minutes and seconds by reference.

Hint: Use the division and modules operators to get the full mark.

Problem 3: (10 points)

The integers (a,b,c) such as (3,4,5) and (5,12,13) are known as "Pythagorean Triples" because they can represent the sides of a right triangle and hence satisfy the Pythagorean theorem:

$$a^2 + b^2 = c^2$$

Given two integers m, n such that $m > n$, a Pythagorean triple can be generated by the following formulas:

$$a = m^2 - n^2, b = 2mn, c = m^2 + n^2$$

Write a function that receives the values for m and n and returns the values of the Pythagorean triples generated by these formulas.

Hint: Use the pow() function to get the full mark.

Problem 4: (10 points)

Write a function **rarray** (x,n) to fill the integer array (x) of size (n) with a random sequence of Zeros (0) and Ones (1).

Hint: Use rand() and srand() to get the full mark.

Problem 5: (10 points)

Write a function **reverse** (x, n) to receive a 1-D array (x) of size (n) and return the array with its elements reversed (e.g. x = {1,5,3,2} returns x = {2,3,5,1}).

Hint: Do not use an extra array to perform the reversal to get the full mark.

Problem 6: (10 points)

Write a boolean function **descendOrder** to receive an array `table[]` of size (n) integers and return true if the values in the array are in descending order and false otherwise.

Hint: Use a single loop only to get the full mark.

Problem 7: (10 points)

- Define an enumerated data type `color` with members `black` , `white` [\[See what an enum is\]](#)
- Declare two 2-D arrays `A[][]` and `B[][]` of type `color` to represent two black and white images having the same size (n x m).
- Implement a Boolean function to receive these two images and return true if A is the negative of B (i.e. each pixel of A is the complement of the corresponding pixel of B) and false otherwise.

Hint: Show examples for true and false cases, to get the full mark.

Problem 8: (10 points)

Implement a Boolean function `Unique (S , Sub)` to test if a string S contains only one copy of substring Sub.

Hint: Show examples for true and false cases to get the full mark.

Problem 9: (10 points)

Assume that the name of a text file is already stored in a C++ string (`infile`). The file contains single words separated by blanks. Write a boolean function `file_ordered (string infile)` to receive the file name string as input parameter and to return true if the words in the file are ordered in ascending alphabetic order and false otherwise. Assume the file to be very long so that you cannot copy it into an array.

Hint: Show examples after and before using the function to get the full mark.

Problem 10: (10 points)

Assume that the name of a file on disk is already stored in a C++ string `infile`. The file consists of a number of lines (a line is a string that might contain blanks and terminated by an end-of-line). Implement a function `maxLen (string infile)` that returns the maximum length of a line in the given file.

Hint: Show examples after and before using the function to get the full mark.

Problem 11: (10 points) + (5 points bonus)

Declare a struct type **point** in 3-D space with coordinates (x,y,z) of type double. Write a function **minDistance** that receives an array of n points in 3-D space and returns the index of the point of minimum distance from the origin (0,0,0).

Hint: Show examples using the function to get the full mark.
