Introduction to Programming

Assignment 03

Due Date

Softcopy submission:- 05-Jan-2023; 23:59 hrs

Hardcopy submission:- 06-Jan-2023; 13:00 hrs

Instructions:

1. Write the assignment on A4 sheets.
2. Write both questions and answers in the assignment. It should be handwritten.
3. Clearly mention your Name, Registration Number, Semester, School and email ID at the beginning of the assignment sheet.
4. After you have completed the assignment, scan the assignment sheets as a single PDF, preferably using Adobe Scanner in mobile, and submit the softcopy of the assignment as PDF in the LMS within the Softcopy Submission deadline.
5. Submit the physical paper copy of the assignment on Hardcopy Submission deadline with multiple papers properly stapled.
6. The assignments will be graded only for those who have submitted both softcopy and hardcopy within the due date.
7. **There will not be any extension to the submission deadline**.

Note:

- For Part-A, it is completely handwritten and elaborate on your answers.
- For Part-B, write the logic or the steps you will use to solve the problem in the assignment sheet.
- You have to submit a working source code for Part-B.
    - No need to write the source code in the assignment sheet.
- So, you will have to make two submissions in LMS
    - The PDF softcopy of the written part
    - C Source code for Part-B as a zip file (all four source codes groped under a folder and then zip the folder)
- If plagiarism is found in both written part as well as the source code, then no marks will be given for this assignment.
- Total marks for this assignment is **40**.

Part A

(Julia Programming)

(10 x 2 = 20)

***Note**:*

1. *Refer to "Think Julia: How to Think Like a Computer Scientist" textbook which is available online at the following link:*

   https://benlauwens.github.io/ThinkJulia.jl/latest/book.html

2. *Write elaborate answers completely justified with reasons/explanations.*

---

1. In a `print` statement, what happens if you leave out one of the parentheses, or both?

2. If you are trying to print a string, what happens if you leave out one of the quotation marks, or both?

3. You can use a minus sign to make a negative number like `-2`. What happens if you put a plus sign before a number? What about `2++2`?

4. In maths notation, leading zeros are okay, as in `02`. What happens if you try this in Julia?

5. What happens if you have two values with no operator between them?

6. We've seen that `n = 42` is legal. What about `42 = n`?

7. How about `x = y = 1`?

8. In some languages every statement ends with a semi-colon, `;`. What happens if you put a semi-colon at the end of a Julia statement?

9. What if you put a `period` at the end of a statement?

10. In maths notation you can multiply `x` and `y` like this: `x y`. What happens if you try that in Julia? What about `5x`?

Part B

(C Programming)

(4 x 5 = 20)

Note:

1. *Write the logic behind how to solve the problems in the assignment sheet.*
2. *The source codes should be submitted as C codes and not in other image formats.*
3. *There should be separate C source codes for each question.*
4. *Put all of them in a single folder and upload the zipped file in LMS.*

---

11. Read an integer array of size `5` from the user. Write a function that takes two arrays as arguments. The first argument is the original array, and the second argument an empty array of size `5`. Inside the function, the empty array should be populated with the index of the original array whose element is an `even` number. The function should keep track of how many even numbers are present in the original array and return that `count`. In the main function, print the second array which holds the indices of the original array.

12. Read two strings `str1` and `str2` from the user. Dynamically allocate 32 bytes using malloc function to `str1` and `str2` . A new string `str3` should concatenate the first-half of `str1` and second-half of `str2`. The size of `str3` should be estimated in runtime and allocated sufficient memory dynamically to hold relevant parts of `str1` and `str2`. The concatenated strings should perfectly reside in `str3` and no memory should be wasted.

Note*: If length is odd, then the first half is integer part of division by 2 and second half is the length minus integer part (i.e, if length is 5, then first half is 2 chars, and second half is 3 chars (i.e., 5-2)).*

Example 1	**Example 2**

Input:
`str1 = abcde`
`str2 = wxyz`
`str3 = abyz`
	Input:
	`str1 = abcd`
	`str2 = vwxyz`
	`str3 = abxyz`

13. Write a recursive function `nr_Digits( )` which returns the number of digits of a given integer number, and also at the same time prints the number in reverse. The number of digits should be printed from the main function, but printing the number in reverse should be handled by the recursive function.

Example
Input: `1234`
Output:
`4321` (printed by the recursive function)
`Length: 4` (printed from the main with the returned value of `nr_Digit( )`)

14. Pass an integer array consisting of five elements to a `void` function named `reverseCubeArray()`. Modify the contents of the array in such a way that the original array is reversed and the contents cubed by making use of a `swapCube()` function called within `reverseCubeArray()`. The `swapCube()` function takes the pointer to two elements which should be swapped so as to reverse the original array with its cube. You may use the `pow( )` function defined in `math.h`

Example
Input: `[1, 2, 3, 4, 5]`
Output: `[125, 64, 27, 8, 1]`