CS 332/532 – 1G- Systems Programming

HW 1

Deadline: 09/15/2024 Sunday 11:59pm

Objectives

Practice C Programming

`primeOrFactorial(n)`

Write the function **`primeOrFactorial`** that takes a **positive integer n** and returns **a string** according to the following conditions:

- If n is a prime number, return `"Prime Number"`.
- If n is not a prime number, return the `factorial of n`.
- If n is less than or equal to 1, return `"Invalid Input"`.

Sample Inputs	Expected Outputs
`n=5`	`"Prime Number"`
`n=4`	`"24"`
`n=1`	`"Invalid Input"`
`n=7`	`"Prime Number"`
`n=-3`	`"Invalid Input"`

`UABIndexSum(arr)`

Write the function **`UABIndexSum`** that takes an **array of integers arr** and returns an **integer**. The function should return the sum of all elements that are equal to their indices. You can pass the size of the array as an input parameter.

Sample Inputs	Expected Outputs
`arr = [0, 2, 2, 4]`	`2`
`arr = [3, 1, 0, 4]`	`0`
`arr = [1, 2, 3, 3]`	`3`

replaceEvenWithZero (arr)

Write the function **`replaceEvenWithZero`** that takes an **array of integers arr** and **returns a new array**. The function should replace every even number in the array with 0. You can pass the size of the array as an input parameter.

Sample Input:	Expected Output:
`arr = [1, 2, 3, 4]`	`[1, 0, 3, 0]`
`arr = [2, 4, 6]`	`[0, 0, 0]`
`arr = [1, 3, 5]`	`[1, 3, 5]`

evenSquare(n)

Write the function `evenSquare` that takes an **integer n** and returns **True** if n is a even square, and **False** otherwise. An even square is a number that can be expressed as the product of an even integer with itself.

Sample Input:	Expected Output
`n = 16`	True
`n = 15`	`False`
`n = 25`	`False`
`n = 36`	`True`

countVowels(s)

Write the function countVowels that takes a string s and returns an integer. The function should count the number of vowels (a, e, i, o, u) in the string, ignoring case.

Sample Input:	Expected Output
`s = "Hello World"`	3
`s = "UAB CS"`	2
`s = "Python"`	1
`s = "aeiou"`	5

Homework Instructions / Grading Rubric

1. Function Implementation

You are required to implement the all five functions. Each function is worth 20 points, for a total of 100 points. Ensure that each function is correctly implemented and passes all test cases provided in the assignment. A problem is correct if all our tests pass, otherwise it will be considered as False.

2. File Structure

Single .c File: All your functions must be written in a single .c file.Name the file `blazerid_HW01.c` Do not submit multiple .c files.(you can create one header file if you wish)
Penalty: 20 points will be deducted if more than one file is submitted.

3. Main Function

Single main Function: Your blazerid_HW01.c file should include a single main function that calls each of the five functions with the sample inputs provided in the assignment. This will demonstrate that your functions work as intended.
Penalty: 20 points will be deducted if more than one main function is included, or if the main function does not call all five functions.

4. General Coding Standards

Follow standard C programming conventions, including proper indentation, variable naming, and commenting where necessary. Ensure your code is readable and free of unnecessary complexity.

Warning

Following instructions properly is a crucial skill and requirement for any computer scientist. Make sure you follow the instructions provided, or you will lose points.
Important: If your code does not compile, you will receive a zero. Ensure that your code compiles and runs correctly on the required platforms. Please refer to **Lab01** for details on the required platforms. In short, your code needs to be compiled on Vulcan server, or you need to demo your code to TA

Submission

- Submit the `blazerid_HW01.c` file to the Canvas.
- The deadline is 09/15/2024 Sunday 11:59pm (*please check the syllabus for the late submission policy*)
- This is **not** a group project, violation of the Academic Integrity Code will result in a failure grade.
- ***** Do not forget to include *"independent completion form"* *****