The assignment will be graded out of 100 points.

Due: 11:59pm Monday, July 9, 2018

Submission Guidelines:

Assignment will be submitted via Blackboard. If there are multiple files that you are submitting then you should zip your files into a single file and submit it. The name of this file should be in following format: lastname_firstname_UTANetID. If you are submitting a single file, make sure it is in the following format: word, txt or pdf and it should be named as mentioned above.

Make sure your name and your student ID are listed in your assignments.

If your assignment is not completed by the deadline, send it anyway and review it with the TA for partial credit. Do not take a zero or excessive late penalties just because it isn't working yet. We will make an effort to grade you on the work you have done.

Note: Please do not forget to zip all the files in ONE folder

Example:

Task1.c
Task2.c
Task3.c
Task4.c
Task5.c
Task6.c
Task7.c

After that, copy all the C files into one folder and compress(zip) the folder. Then, submit the compressed (zipped) folder on Blackboard.

Assignment Specification:

For all the tasks, get the input from the user.

Task 1 (10 pts.)

In a file called task1.c, write a C program to compute Fibonacci(n) given N using **recursion** (Recursive Function).

Example:

Input

Enter the nth number in fibonacci series: 8

Expected Output:

The 8th number in fibonacci series is 21

Task 2 (15 pts.)

In a file called task2.c, write a C program to find the sum of digits of a number using **recursion** (Recursive Function).

Example:

Input:

Enter a number to find sum of digits: 376

Expected Output: The Sum of digits of 376 = 16

Task 3 (15 pts.)

In a file called task3.c, write a C program to calculate the Greatest common divisor (GCD) of Two Numbers using **Recursion** (Recursive Function).

Example:

Input:

Enter two positive integers: 525 50

Expected Output:

GCD of 525 and 50 is 25

Task 4 (15 pts.)

In a file called task4.c, write a C program to print the following series: 1 2 5 26 677 ..... such that the nth value equals to (n-1 th)^2 +1 and the first value is 1. Write a recursion function named myFunc to compute the nth value. Use for loop to print the first 6 values.

Task 5 (15 pts.)

In a file called task5.c, write a C program to find whether a Number is Prime or Not, using **Recursion** (Recursive Function).

Hint: A Prime Number can be divided evenly only by 1, or itself. And it must be a whole number greater than 1.
Example: 5 can only be divided evenly by 1 or 5, so it is a prime number. But 6 can be divided evenly by 1, 2, 3 and 6 so it is NOT a prime number.

Example:

Input:
Enter a positive integer: 3
Expected output: 3 is a prime number.

Another example:
Input:
Enter a positive integer: **14**
Expected output: 14 is NOT a prime number.

Task 6 (15 pts.)

In a file called task6.c, write a C program to compute the average of 3 numbers (write a void function and call that function **by reference, not by value)**. Hint: you can use an array to pass it to a function.

Example:

Input:

Enter 3 integer numbers: 5 7 9

Expected output:

The average of the three numbers is: 7

Task 7 (15 pts.)

In a file called task7.c, write a C program to reverse a string using recursion (recursive function).

Example:

Input: Programming
Expected Output: gnimmargorP

Another Example:
Input: CSE1320
Expected output: 0231ESC

Assignment Guidelines:

There will be several programming assignments in this course, typically assigned on a weekly basis. All assignments will have equal weight. No assignment scores will be dropped. The following class policies regarding assignments will be followed:

- All assignments must be submitted via Blackboard.
- No deadline extensions for the entire class will be provided. (See syllabus about policy on extensions for individuals, based on emergencies documented in writing).
- No extra credit will be provided.
- If you make multiple submissions to Blackboard for the same assignment, only the latest submission will be graded.

Late submission policy:

- All assignments are graded out of 100 points. Assignments submitted late will be penalized, at a rate of 4 penalty points per hour. The submission time will be the time shown on Blackboard. Any assignment submitted more than 25 hours late will receive no credit.
- Exceptions to late submission penalties will only be made for emergencies documented in writing, in strict adherence to UTA policy. For all such exception requests, the student must demonstrate that he or she made all efforts to notify the instructor as early as possible.
- Computer crashes, network crashes, and software or hardware failure will NOT be accepted as justification for late submissions. If you want to minimize chances of a late submission, aim to submit early. You can always revise your submission till the deadline.
- Sometimes students submit the wrong files on Blackboard. Unfortunately, no credit or waiver of late penalties can be provided in such cases.
- If you find yourself in an emergency situation and cannot deliver homework on time, immediately inform the instructor and teaching assistant. Even if you have a valid reason for delivering late an assignment, you must make a convincing case that you have notified the instructor and teaching assistant as early as possible.

If you want to minimize chances of a late submission, aim to submit early. You can always revise your submission till the deadline (maximum 3 attempts).