Object-Oriented Programming Lab

SPRING - 2023

LAB 05

FAST National University of Computer and Emerging Sciences

Learning Outcomes

In this lab you are expected to learn the following:

- Recursion

Lab Tasks

Submission Instructions:

1. Create a single cpp file containing all the functions of the problems and main function.
2. Save the **cpp** file with the task number e.g. Q1.cpp
3. Now create a new folder with name *ROLLNO_SEC_LAB01* **e.g. i22XXXX_A_LAB05**
4. You need to display your roll no and name before the output of each question.
5. Move all of your .**cpp files (without the main function i.e., comment out the mainfunction)** to this newly created directory and compress it into a **.zip file**.
6. Now you have to submit this zipped file on Google Classroom.
7. If you don't follow the above-mentioned submission instruction, you will be marked **zero.**
8. Plagiarism in the Lab Task will result in **zero** marks in the whole category.

Q 1. Given an array A, make a triangle such that the bottom most level has all array elements. Then, at each level number of elements is one more than the previous level. Elements at each level is the sum of consecutive elements in the previous level.

Example:

Input: [1,2,3,4,5]

Output: [48]

[20,28]

[8,12,16]

[3, 5, 7, 9]

Prototype:
int** sum_of_sequence(int *p, int size)

Q 2. Print all numbers in range [100,999] such that sum of digits at even position is equal to sum of digits at odd position.

Output:

110, 121, 132, 143, 154, 165, 176, 187, 198, 220, 231, 242, 253, 264, 275, 286, 297, 330, 341, 352, 363, 374, 385, 296, 440, 451, 462, 473, 484, 495, 550, 561, 572, 583, 594, 660, 671, 682, 693, 770, 781, 792, 880, 891, 990

Prototype:
int* sum_of_digits(int start, int end ,int size)

Q 3. Write a recursive function that calculates the greatest common divisor (GCD) of a given number. The **GCD** of two numbers is the largest number that divides them both.

Example:

30: 1,2,3,5,6,10,15,30

24: 1,2,4,6,12,24

GCD=6

Prototype:
int GCD(int num1,int num2)

Q 3. Write a C++ recursive function to print following pattern**.**

Prototype:
void diamond_pattern(int num)

For n=13 , n=11 , n=9 , n=7