Assignment -1

CAP222J: COMPUTER APPLICATIONS (PROGRAMMING FUNDAMENTALS THROUGH C: UG 2nd Semester

Due Date: Monday, 29th April, 2024

Marks: 10

This assignment aims to test your understanding of basic programming concepts in C, including loops, conditional statements, and mathematical operations. You are required to write three separate C programs, as detailed below:

Program 1: Armstrong Numbers

Write a C program that identifies and prints all Armstrong numbers between 1 and 1000. An Armstrong number is a number where the sum of the cubes of its digits is equal to the number itself. For example, 153 is an Armstrong number because 1^3 + 5^3 + 3^3 = 153.

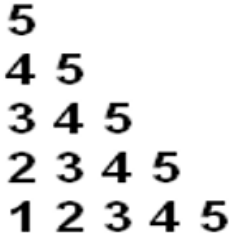
Program 2: Reverse Number

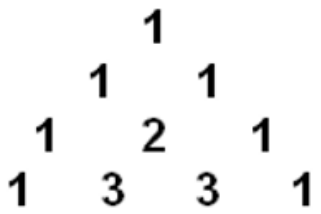
Write a C program that takes an integer as input and prints the number in reverse order. For example, if the input is 1234, the output should be 4321.

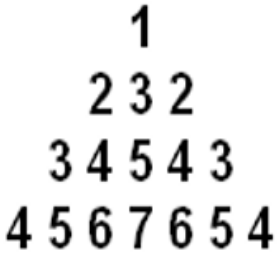
Program 3: Factorial Calculation

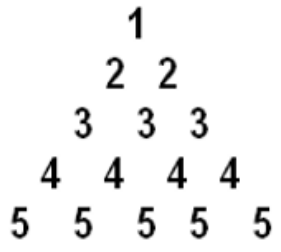
Write a C program that calculates the factorial of a given non-negative integer. The factorial of a number is the product of all positive integers less than or equal to that number. For example, the factorial of 5 (denoted as 5!) is 5 x 4 x 3 x 2 x 1 = 120.

Program 4-9: Write a program to print the following Outputs:









Submission Instructions:

1. Write the code for all the programs and ensure they run correctly on your computer.
2. Print a hard copy of each program's code.
3. Demonstrate the execution of each program to your instructor or teaching assistant.
4. Submit the hard copies of your code by the due date.

Grading Rubric:

Your assignment will be evaluated based on the following criteria:

1. Correctness: The programs should generate the correct output for all valid inputs.
2. Efficiency: The code should be efficient and avoid unnecessary computations.
3. Readability: The code should be well-structured, properly indented, and use meaningful variable names.
4. Comments: The code should include comments explaining the logic and functionality of different sections.

Tips:

1. Start by planning the logic of each program before writing the code.
2. Use appropriate data types for variables.
3. Break down the problem into smaller functions to improve modularity and readability.
4. Test your code with various inputs to ensure it works correctly in all cases.

Good luck!

Teacher In charge:

Prof. Muhammad Iqbal Bhat
Government Degree College Beerwah