**Assignment 2**
**Topics**: Expressions and operators

Section A2.1: [Arithmetic Operators and Expressions]

- A2.1a: Write a C program to perform the addition, subtraction, multiplication, and division of two numbers entered by the user. Display the results of each operation.

- A2.1b: Create a program to calculate the square and cube of a number entered by the user using multiplication ('*').

- A2.1c: Implement a program that calculates the average of three numbers entered by the user. The result should display both the sum and the average.

- A2.1d: [Bonus] Write a program to solve the expression '((a + b) * c - d) / e' with user inputs for variables 'a, b, c, d, e'. Ensure the program checks for division by zero and handles it gracefully.

- A2.1e: Write a program to evaluate the expression 'a + b * c / d - e' for user input values of 'a, b, c, d,' and 'e'. Explain the order of operations and precedence used to obtain the result.

---

Section A2.2: [Relational and Logical Operators]

- A2.2a: Write a C program to check if a given number is even or odd using the modulus operator ('%').

- A2.2b: Create a program that takes three numbers as input and checks if the first number is greater than the second and less than the third using relational operators.

- A2.2c: Develop a program that accepts three angles of a triangle as input and checks if the triangle is valid or not. (A triangle is valid if the sum of all three angles is equal to 180 degrees.)

- A2.2d: [Bonus] Write a program to determine if a number entered by the user is within a range (e.g., between 10 and 100) using logical operators ('&&', '||'). Additionally, implement a feature to check if the number is either less than 10 or greater than 100.

---

Section A2.3: [Assignment Operators and Increments/Decrements]

- A2.3a: Write a program to demonstrate the use of pre-increment ('++a') and post-increment ('a++'). Show the difference between them by printing their values before and after the increment.

- A2.3b: Create a program that uses the ternary operator to find and print the smaller of two numbers entered by the user.

- A2.3c: Write a program that uses combined assignment operators ('+=', '-=', '*=', '/=') to update a variable multiple times based on user input.

- A2.3d: [Bonus] Implement a program that calculates the factorial of a number using a for-loop and demonstrates the use of both pre-decrement ('--i') and post-decrement ('i--').

---

Section A2.4: [Bitwise, 'sizeof', and Address-of Operators]

- A2.4a: [**Optional**] Write a C program that demonstrates the use of bitwise AND ('&'), OR ('|'), and XOR ('^') operators on two user-input integers. Display the binary representation and the results of each operation.

- A2.4b: Create a program to show the use of combined assignment operators ('+=', '-=', '*=') by taking an initial value and performing multiple operations on it.

- A2.4c: [**Optional**] Write a program to toggle the case of a given character using bitwise XOR. If the character is lowercase, convert it to uppercase, and vice versa.

- A2.4d: Write a program that uses the 'sizeof' operator to display the sizes (in bytes) of various data types ('int', 'float', 'char', 'double', etc.). Explain the output.

- A2.4e: Develop a program to demonstrate the use of the address-of operator ('&'). Declare a variable, print its value, and then print its memory address using the '&' operator. Then use a pointer to print the value stored at the address.

- A2.4f: Write a program that rotates the bits of an integer left by 'n' positions, where 'n' is provided by the user. Use bitwise operators to implement the rotation and print the result in binary form.

---

Section A2.5: [Type Conversion and Typecasting]

- A2.5a: Write a program that demonstrates implicit type conversion by adding an integer and a floating-point number. Print both the input values and the result.

- A2.5b: Create a program that explicitly converts a floating-point number to an integer using typecasting and displays both the original float value and the converted integer value.

- A2.5c: Write a program to calculate the area of a circle. The radius should be input as an integer, but calculations should be performed in floating-point arithmetic. Use typecasting to ensure precision in the final result.

- A2.5d: [Bonus] Implement a calculator program that accepts two inputs of different types (e.g., an 'int' and a 'float'). Perform addition, subtraction, multiplication, and division using typecasting to ensure that the operations are conducted in floating-point arithmetic when necessary.

- A2.5e: Write a program to demonstrate typecasting by converting an integer division to a floating-point division. Accept two integers from the user and display the result of the division both with and without typecasting.

---