Programming Fundamentals (CS1002) | Fall 2023 | BCS-1A

Assignment No. 02 Total marks:

Instructions:

1. Read the problem statement carefully and understand what is being asked. Failure to comprehend the problem statement may result in incorrect assumptions and answers.
2. **Plagiarism will not be tolerated. Any evidence of copied work, including minor instances, will result in a grade of zero for the entire assignment.**
3. **Do not use logical operators and loops. Solve using function arguments, if-else statements, and function calls.**
4. Late submissions will not be accepted under any circumstances.
5. Submit the PDF file to Google Classroom. Upload the
6. Name your PDF file using the following convention: ROLLNO-NAME, e.g. (23P-8743-Zain.pdf).
7. Assignment should be submitted in a Zip file named **2XP_XXXX_Name.zip**
8. Create a PDF file with screenshots of the code and output. The zip file should contain **PDF** and all **.c files**.
9. Create a Name .c file like **2XP_XXXX_Name_QX.c**.
10. The code must contain proper indentation and comments, or else marks will be deducted.
11. In case of an invalid input, your function shall return -1.
12. If you have any questions or concerns, ask before submitting your work. Lame excuses or attempts to justify plagiarism or late submissions will not be accepted.
13. **If the above instructions are not followed, marks will be deducted, and the assignment deadline is not extendable.**

Question No. 01:

In a futuristic city in the year 3077, there are two modes of travel between different dimensions: **teleportation portals** and **spaceships**.

The city council is interested in calculating the total expense of transportation for a specific number of teleportation portals and spaceships. They want to determine the cost of each mode of transportation and aggregate it.

Using the **Modular programming approach**, Define a function for each calculation and give meaningful names to the functions and variables that represent the working. Functions will take teleportation portals and spaceships **count**, and the distance for spaceships as arguments and return the cost of each mode of transportation.

Take the **hypothetical** cost of a portal and spaceship and its cost per distance peruse. Use the preferred macro to define the cost per use of a teleportation portal and a spaceship and the cost per distance. This is a good practice, as it makes the code easier to maintain, and the cost values cannot be changed throughout the program. The function return values must be protected so they cannot be changed in the calling function.

The program must have a function to calculate the total transportation cost, further calling the functions needed. This function will take the number of travel modes as arguments. It will use the parameter functions to calculate the cost of each mode of transportation.

Question No. 02:

In this world where cars are powered by mathematical formulas. A team of engineers is tasked with creating a new engine that can provide more power and efficiency. The team is given two mathematical functions, **formulaOne** and **formulaTwo**. The function **formulaOne** takes two integers as input, representing the power of the car's engine and the speed at which the car is moving. It returns their sum, which represents the **total power output** of the car. The function **formulaTwo** takes a single integer as input, representing the efficiency of the car's engine. It returns the square of this integer, which represents the **engine's efficiency at high speeds**.

The engineers must use these functions to calculate the optimal power output of the car. They must find the sum of the total power output and the square of the efficiency at high speeds, which will determine the overall performance of the car. To achieve this, you should write a statement that calculates the total sum of the

optimal powers of two given cars and provides the result in the form of a new car engine's overall performance,

Calculate the new car's overall performance without using any logical operators, loops, arrays, strings, or recursion.

INPUT:

Car 1 power = 10
Car 1 speed = 20
Car 1 efficiency = 40

Car 2 power = 30
Car 2 speed = 50
Car 2 efficiency = 60

OUTPUT:

New cars overall performance: 5310

Hint: T**he optimal power of a car is equal to the overall performance**,

---

Question No. 03:

You are working on a project that involves calculating various metrics for a company's financial performance. You have been provided with a set of data that includes revenue, expenses, profits, and other relevant financial figures for the year. Your task is to write a program that calculates and displays the following metrics:

- The total revenue for the year.
- The total expenses for the year.
- The net profit (profit = revenue - expenses) for the year.
- The profit margin (profit margin = net profit / revenue) for the year.
- The return on investment (ROI = net profit / expenses) for the year.

Question:

Write a C program that calculates and displays the financial metrics for a company. Your program should be able to handle the following scenarios:

- If the user enters input that triggers mathematical errors (**division by zero**), the program should handle these cases by returning **-1.0** and providing informative error messages. If the profit margin and ROI are not calculated properly due to error, return **1** to exit the program
- Use data structure with a decimal point for precise calculations.

Question No. 04:

You are working on a project that involves placing boxes of different colors over each other. The goal is to place boxes, but the rule is that only a box of different color can be placed over a box of particular color. Your task is to write a program that checks if the placement of a box is possible or not.

Return **1** valid placement and **-1** for invalid placement from the function, otherwise you could use bool as a return type using the library `#include <stdbool.h>.`

Question:

Write a C program that checks If the boxes can be placed in a valid placement. Your program should be able to handle the following scenarios:

- Take two boxes from the user from these two colors, **red , blue or green** . Check if a box can be put over the other one, and show a message according to the valid or invalid placement.
