COMP1236 Group Project

Use the Bash shell for the completion of this project.

Develop a shell scripting application that allows the user to perform some advanced mathematical operations.

You application should allow the user to perform these three tasks:

- Task 1: Find the even multiples of any number chosen by the user in a given range – the user should specify the multiples of which number they want to print and a minimum and maximum value. The multiples printed should be within the [minimum, maximum] range. For example, if the user choses the number 7 and would like to print all even multiples in the [15, 60] range, the program should print all the even multiples that are bigger than or equal to 15 (minimum), but smaller than or equal to 60 (maximum). The program should keep a count of how many numbers were printed and should output that count, as well as the numbers. The output then, for the example given, should be 28, 42, 56, and the count should be 3.

- Task 2: Find the terms of any linear sequence given by the rule Term = a*n + b, where a and b are integers specified by the user and n is a positive integer and print them in order (for example if the user inputs a=3, b=-4, the first few terms that should be printed are -1, 2, 5, 8, 11…). The user also will specify how many terms the program should print. In addition, the program should print the sum of terms found and a count of how many odd terms were found.

- Task 3: Find the numbers that can be expressed as the product of two nonnegative integers in succession and print them in increasing order. For each number found the program should check whether the number is a multiple of 5 and indicate this in the output. (For example, 30 is such a number as it can be written as the product of 5 and 6 (30=5x6), which are two numbers in succession.) The user should specify how many such numbers they want to print.

Your application should be user friendly. First, you create an initial menu that the user sees when they execute the application. The menu should display the list of operations (the three tasks) that the application can perform.

In your menu design, if user enters M or m, the application should print a list of the multiples of the number (Task 1). The user determines a minimum and a maximum value too.

If the user enters S or s, the application should print a list of the terms for the chosen sequence, a count of how many odd terms were found and the sum of the terms (Task 2). The user determines how many terms should be printed.

If the user enters P or p, the program should print a list of the numbers which are products of two numbers in succession, and whether each number found is a multiple of 5 (Task 3). The user specifies how many numbers to print.

The program should display the user's selection before printing the output. If the user enters a selection which is different from the choices offered, the program should output an error message. After a task has been completed, the user should be given the option to select another task, or to exit.

Before writing the code, you should work on the problem solving part of the application development process. Draw the flowcharts for each of the tasks separately. Next, you need to draw the main flowchart which will show the logic of your entire application.

Marks: 20% of course grade

Task	**Possible marks**	**Description**
Problem Solving - Flowcharts	15	The flowchart shows the logic of your program correctly
Application Menu Interaction	10	The application menu displays correctly. This includes each task's interaction with the user while getting the input
Task 1 Functionality	15	The application displays the required multiples correctly
Task 2 Functionality	15	The application displays the required sequence numbers correctly
Task 3 Functionality	15	The application displays the required amount of numbers correctly
Application Functionality	20	The user can switch between tasks and everything works correctly as a whole.
Code Documentation and Conventions	10	Comments are entered appropriately and best conventions discussed in class are followed.

Submission guideliness:

Complete this project by Sunday, 12th of December.

You should submit a zip file named **Group_No_Project.zip** which contains three files:

A single pdf file which contains the flowchart(s) named: **Group_No_Flowchart.pdf**

A single .sh file which contains your code named: **Group_No_Application.sh**

A single document where you have indicated the software used for completion of your tasks and how the work was distributed amongst team members.