SECTION D: REPETITION/ITERATION

For Loop

1. Write a program **Addition** that asks the user how many number he wants to add. The user is then prompted to enter the numbers and the program finds and displays the total.

2. Write a program **Countdown** which simulates the countdown for midnight starting from the last 10 seconds. Show each number on its own on screen.

3. Write a program **Factorial_5** which calculates the factorial of 5.

4. Write a program **Numbers** which generates 30 random numbers between 1 and 5 and displays them 10 in a row. At the end, the program displays how many times each number was generated.

5. Write a program **Highest** which asks the user to input 10 numbers. The average of these numbers is calculated along with the largest number. Both the average and largest number are to be displayed.

6. Write a program **naturalNumbers** to display the n terms of natural numbers and their sum. For example, if 5 is entered, the natural numbers are 1, 2, 3, 4 and 5 while the sum is 15.

7. Create a class **nameCode** which asks the user to input every letter of a name, saved in separate variables. The program should then display the name of the person in ASCII.

While Loop

8. Write a program **rangeMarks** which continually asks the user to input a mark between 0 and 100 inclusive until the user has entered a valid mark.
9. Write a program **numberGuess** which generates a whole random number between 1 and 10. The user is then asked to guess the number. The program stops when the user has guessed the number and the number of guesses is also displayed.

10. Write a program **Odd** which displays all the odd numbers between 1 and 100.

11. Modify the class **Lock** in Section C so that the user is given three chances to open the lock. Another chance to input the combination is given only if the previous combination was wrong and the three chances have not been used up.

Do While Loop

12. Write a program **avgNumbers** that asks the user to input numbers and only stops when the user inputs 0. The program finds the average of the list of numbers entered and displays it to the user.

13. Write a program **oddNumbers** that asks the user to input a list of 20 whole numbers. The total of how many odd numbers were entered is then displayed.
14. Write a program **Marks** which asks the user to input the pass mark of an exam. The program then asks the user to input 10 student names and marks. The program must show whether each student has passed or failed. Finally, the program displays the number of students who have passed and the number of students who have failed. The name of the student with the highest marks is also shown along with the marks obtained.
15. Write a program **Shapes** which shows the following menu to the user:

    1. Rectangle
    2. Square
    3. Quit

    Once the user chooses an option, the appropriate sub-menu is chosen. Both Option 1 and Option 2 have a further two options: Area and Perimeter. Ensure that each option works and cater for invalid data.

16. Write a program **Letter** which generates a random letter between A and Z. The user is asked to guess the letter. A message “Too low” or “Too high” should be displayed after each guess. The user has 3 tries to guess the letter. Each time the user guesses incorrectly, the number of tries left is to be displayed. The user can decide to give up by entering *. In this case the random letter should be displayed to the user.

Nested Loop

17. Write a program **Factorial** that finds the factorial of the numbers 1 through 10.
18. Write a program **Triangle** that outputs the following pattern:

```
***
**
*
```

19. A prime number is a number that is only divisible by itself and 1. Write a program **primeNumbers** which finds the prime numbers up to 50.

20. Create a class called **Factors** which finds the factors of the numbers from 2 to 20. Nested loops are needed to solve this problem.