Day 1 - Assignments

Pseudo Code

- Check whether a number is even or odd

```
If the number % 2==0
        Then number is even
Else
        Number is odd
```

- Find the largest of three numbers

```
If num1>num2 and num1>num3
        Then num1 greater
Else if num2>num1 and num2>num3
        Then num2 greater
Else
        num3 greater
```

- Display the multiplication table for any number

```
Loop from 1 to 10 as variable i
        Print number x i
```

- Calculate the sum of first N natural numbers

```
sum=(N*(N+1))/2
```

- Find the factorial of a number

```
Defina a Function factorial(num)
        If num==1 or num==0
                Then Return 1
        Else
                Return factorial (num-1)*num
```

- Calculate the average and grade of 5 subject marks

```
Mark_sum = mark1+mark2+mark3+mark4+mark5
Average = Mark_sum/5
Define Function Find_grade(mark)
        If mark>90
                Then Print ‘a’
        If mark>80
                Then Print ‘b’
        If mark>70
```

```
                Then Print ‘c’
        If mark>60
                Then Print ‘d’
        If mark>50
                Then Print ‘fail’

Call Find_grade(mark1)
Call Find_grade(mark2)
Call Find_grade(mark3)
Call Find_grade(mark4)
Call Find_grade(mark5)
```

- Find the largest and smallest element in an array

```
Sort the array
Smallest element =array[0]
Largest element =array[size(array)-1]
```

- Count how many even and odd numbers are in a list

```
For number in list
        If number % 2==0
                Then even_count=even_count+1
        Else
                odd_count=odd_count+1

Print even_count
Print odd_count
```

- Generate a pattern like a pyramid or triangle

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

```
Loop form 1 to row count as i
        Loop from 1 to i including i
                Print “* ” in the sameline
        Print newline
```

- Find the second largest number in a list

```
sort(list)
reverse(list)
set(list) to remove duplicate values
Print list[1]
```

- Find the sum of diagonal elements in a 2D matrix

```
Matrix [row][col]
Loop from i=0 to col
    For each iteration
        Add matrix[i][i] to diagonal sum
        Add matrix[i][col-1-i] to diagonal sum

Print diagonal sum
```