**Lab Manual – Mul, Div Instructions and Subroutines**

**Note: In following functions use appropriate local variables where required.**

**Activity 1:** Write a subroutine **Power** that takes x and y as parameters, calculates **x** raise to power **y** and returns the result. Use MUL instruction for multiplication and test your function with three different data/function calls.

**Activity 2:** Write a subroutine **SeriesSum** that calculate the following sum:

Where r and l are variables to be passed through stack. Use Power function (written in activity 1) to calculate r^n. Parameter passing from one subroutine to the other should be via stack. The final answer should be returned from stack as well. For example, if r =2 and l = 8 then final answer is 1FE.

**Activity 3:** Write a subroutine **BinarySearch** that takes (base address of) a sorted array and its size and a key as parameters, searches the key from array using binary search. If the element is found set AX to one and otherwise to zero. Use DIV instruction to find middle index in each iteration. Test your function on three different arrays and keys (three function calls).

**Practice Problem (Exercises from Book):**

1. Write a sub-routine **Factorial** that take a number as parameter, calculates its factorial (use MUL instruction for multiplication) and returns the result. Test your function on three different inputs (with three function calls).
2. Write the sub-routine to calculate factorial. The sub-routine should take as parameter the number to calculate the factorial of and returns factorial through stack. Write a multiplication subroutine instead of using MUL instruction.
3. Declare a 32byte buffer containing random data. Consider for this problem that the bits in these 32 bytes are numbered from 0 to 255. Declare another byte that contains the starting bit number. Write a program to copy the byte starting at this starting bit number in the AX register. Be careful that the starting bit number may not be a multiple of 8 and therefore the bits of the desired byte will be split into two bytes.