

LAB IV

1. Login to default directory and see if the directory **LAB4** exists. If the directory **LAB4** exists, then remove it. Now create the directory **LAB4** and go to the directory **LAB4**. Create all the programs in the directory **LAB4**
2. Write a C program that calculates the sum of integers between 9 and 300 inclusive which are divisible by 7 but not divisible by 63.

Expected output:

```
Sum of integers between 9 & 300 that are divisible by 7 but not by 63 is 5684
```

3. Write a C program that accepts an integer from keyboard and calculates the sum of digits of an integer.

Test data and expected output:

```
Enter an integer: 3456
Sum of the digits of 3456 is 18
```

```
Enter an integer: -23
Sum of the digits of -23 is 5
```

4. Write a C program that accepts a positive integer n less than 50 from the terminal and prints out the sum $1^4 + 2^4 + 4^4 + 7^4 + 11^4 + \dots + m^4$, where m is less than or equal to n . If the input is outside the range, the program terminates with appropriate message.

Test data and expected output:

```
Enter a +ve integer less than 50: 0
Invalid input
```

```
Enter a +ve integer less than 50: 39
Sum of the series is 2898549
```

```
Enter a +ve integer less than 50: 0
Invalid input
```

5. Write a C program that asks the user to enter a positive integer n less than 10. If the user enters an invalid input, the code repeats the command of asking the user for a positive integer less than 10 until the input is correct. It then prints out the sum of the first n terms of the series $1^4 + 2^4 + 4^4 + 7^4 + 11^4 + \dots$.

Test data and expected output:

```
Enter a +ve integer less than 10: 0
Invalid input, enter again: 4
Sum of the 4 terms of the series is 2674
```

```
Enter a +ve integer less than 10: 4
Sum of the 4 terms of the series is 2674
```

```
Enter a +ve integer less than 10: 11
Invalid input, enter again: 5
Sum of the 5 terms of the series is 17315
```

6. Write a C program that accepts integers from the keyboard until we enter a zero or a negative number. The program will output the number of positive values entered, the minimum value, the maximum value and the average of all numbers.

Test data and expected output:

```
Enter a +ve integer:6
Enter next +ve integer:3
Enter next +ve integer:2
Enter next +ve integer:-4
Number of +ve values entered is 3
Maximum value entered is 6
Minimum value entered is 2
Average value is 3.6667
```

```
Enter a +ve integer:0
No positive number entered
```

7. Write a C program that reads an integer n from the keyboard and prints out the factorial of n .

Test data and expected output:

```
Enter an integer:6
Factorial of 6 is 720
```

```
Enter an integer:-3
n must be non-negative
```

8. Write a C program that accepts a positive integer n and a real number x from the keyboard and prints out the sum of the n terms of the series

$$\sin(x) = \sum_{n=0}^{\infty} t_n, \quad t_n = (-1)^n \frac{x^{2n+1}}{(2n+1)!}$$

Test data and expected output:

```
Enter the value of n & x:0 1.0
Number of terms must be +ve
```

```
Enter the value of n & x:5 0.5
Sum of the series at x=0.50 with 5 terms is 0.47943
```

9. Write a C program that reads a real number x from the keyboard and calculates the sum of the series

$$\sin(x) = \sum_{n=0}^{\infty} t_n, \quad t_n = (-1)^n \frac{x^{2n+1}}{(2n+1)!}$$

by adding terms as long as $|t_n| > 10^{-6}$.

Test data and expected output:

```
Enter the value of x:0.8
Sum of the series at 0.80 is 0.71736
```

10. Write a C program that accepts a non-negative integer from the keyboard and checks whether the entered number is a palindrome number.

Test data and expected output:

Enter a non-negative integer:9
9 is a palindrome number

Enter a non-negative integer:246642
246642 is a palindrome number

Enter a non-negative integer:24312
24312 is NOT a palindrome number

11. Write a C program that accepts a non-negative integer from the keyboard and checks whether the entered number is a Armstrong number. (An n -digit number that is the sum of the n -th powers of its digits is called an Armstrong number)

Test data and expected output:

Enter a non-negative integer:-2
Input must be non-negative integer

Enter a non-negative integer:9
9 is an Armstrong number

Enter a non-negative integer:1634
1634 is an Armstrong number

Enter a non-negative integer:1636
1636 is NOT an Armstrong number

12. A perfect number is a positive number in which sum of all positive divisors excluding that number is equal to that number. Write a C program that accepts a positive integer from the keyboard and checks whether the entered number is a perfect number.

Test data and expected output:

Enter a positive integer:-2
Input must be positive

Enter a positive integer:8128
8128 is a perfect number

Enter a positive integer:28
28 is a perfect number

Enter a positive integer:64
64 is NOT a perfect number

13. Write a C program that prints out the prime numbers between 1 and 100. The output should be such that each row contains a maximum of 7 prime numbers.

Expected output:

The prime numbers between 1 and 99 are:

| | | | | | | |
|----|----|----|----|----|----|----|
| 2 | 3 | 5 | 7 | 11 | 13 | 17 |
| 19 | 23 | 29 | 31 | 37 | 41 | 43 |
| 47 | 53 | 59 | 61 | 67 | 71 | 73 |
| 79 | 83 | 89 | 97 | | | |

14. Write a C program that reads a real number x from the keyboard and calculates the sum of the series

$$\exp(x) = \sum_{n=0}^{\infty} t_n, \quad t_n = \frac{x^n}{n!}$$

by adding terms as long as $|t_n| > 10^{-8}$. Also, print out the value from the C math library function $\exp(x)$.

Test data and expected output:

```
Enter the value of x:1.5
Sum of the series at 1.50 is 4.48169e+00
Value from C math library=4.48169e+00
```

```
Enter the value of x:-20
Sum of the series at -20.00 is 2.06115e-09
Value from C math library=2.06115e-09
```