**Experiment 6: To implement Langrange’s Interpolation formula.**

PS C:\Users\admin\Documents\Saksham Gupta\MSIT> ./a.exe

Enter the number of set of values of x and y: 4

Enter the value of x0 :0

Enter the value of y0 :2

Enter the value of x1 :1

Enter the value of y1 :3

Enter the value of x2 :2

Enter the value of y2 :12

Enter the value of x3 :5

Enter the value of y3 :147

Enter the value of x for which y to be found out :3

Y for the value 3 is 35

Figure 6.1: Output of the program

**Experiment 7: To implement Newton’s Divided Difference formula.**

PS C:\Users\admin\Documents\Saksham Gupta\MSIT> ./a.exe

Enter the number of set of values of x and y: 4

Enter the value of x0 :1

Enter the value of y0 :14

Enter the value of x1 :2

Enter the value of y1 :15

Enter the value of x2 :4

Enter the value of y2 :5

Enter the value of x3 :6

Enter the value of y3 :9

Enter the value of x for which y to be found out :5

Y for the value 5 is 3

Figure 7.1: Output of the program

**Experiment 8: Program for solving numerical integration by Trapezoidal rule**

PS C:\Users\admin\Documents\Saksham Gupta\MSIT> ./a.exe

Enter the lower bound of integration for the function:0

Enter the upper bound of integration for the function:6

Enter the number of widths for the function:6

The integral of the function is: 1.41079855

PS C:\Users\admin\Documents\Saksham Gupta\MSIT> ./a.exe

Enter the lower bound of integration for the function:0

Enter the upper bound of integration for the function:6

Enter the number of widths for the function:100

The integral of the function is: 1.40564525

PS C:\Users\admin\Documents\Saksham Gupta\MSIT> ./a.exe

Enter the lower bound of integration for the function:0

Enter the upper bound of integration for the function:6

Enter the number of widths for the function:1000

The integral of the function is: 1.40564716

Figure 8.1: Output of the program

**Experiment 9: Program for solving numerical integration by Simpson’s 1/3 rule**

PS C:\Users\admin\Documents\Saksham Gupta\MSIT> ./a.exe

Enter the lower bound of integration for the function:0

Enter the upper bound of integration for the function:0.6

Enter the number of widths for the function (n):6

The integral of the function is: 0.535155654

PS C:\Users\admin\Documents\Saksham Gupta\MSIT> ./a.exe

Enter the lower bound of integration for the function:0

Enter the upper bound of integration for the function:0.6

Enter the number of widths for the function (n):95

Since n is not even taking n as: 96

The integral of the function is: 0.535153508

Figure 9.1: Output of the program

**Experiment 10: To implement Numerical Integration Simpson 3/8 rule.**

PS C:\Users\admin\Documents\Saksham Gupta\MSIT> ./a.exe

Enter the lower bound of integration for the function:0

Enter the upper bound of integration for the function:0.6

Enter the number of widths for the function (n):6

The integral of the function is: 0.535158396

PS C:\Users\admin\Documents\Saksham Gupta\MSIT> ./a.exe

Enter the lower bound of integration for the function:0

Enter the upper bound of integration for the function:0.6

Enter the number of widths for the function (n):95

Since n is not even taking n as: 96

The integral of the function is: 0.535153568

Figure 10.1: Output of the program

**Experiment 11: Inverse of a system of linear equations using Gauss-Jordan method.**

PS C:\Users\admin\Documents\Saksham Gupta\MSIT> ./a.exe

Input data for Matrix:

Enter the order of square matrix: 2

Enter the number at position (0,0): 1

Enter the number at position (0,1): 3

Enter the number at position (1,0): 2

Enter the number at position (1,1): 5

A Inverse:

-5 3

2 -1

Figure 11.1: Output of the pass case of 2x2 matrix

PS C:\Users\admin\Documents\Saksham Gupta\MSIT> ./a.exe

Input data for Matrix:

Enter the order of square matrix: 3

Enter the number at position (0,0): 5

Enter the number at position (0,1): 7

Enter the number at position (0,2): 9

Enter the number at position (1,0): 4

Enter the number at position (1,1): 3

Enter the number at position (1,2): 8

Enter the number at position (2,0): 7

Enter the number at position (2,1): 5

Enter the number at position (2,2): 6

A Inverse:

-0.209524 0.0285715 0.27619

0.304762 -0.314286 -0.0380952

-0.00952383 0.228571 -0.12381

Figure 11.2: Output of the pass case of 3x3 matrix

PS C:\Users\admin\Documents\Saksham Gupta\MSIT> ./a.exe

Input data for Matrix:

Enter the order of square matrix: 2

Enter the number at position (0,0): 1

Enter the number at position (0,1): 1

Enter the number at position (1,0): 2

Enter the number at position (1,1): 2

A Inverse:

inf -inf

-inf inf

Figure 11.3: Output of the fail case

**Experiment 12: Find the Eigen values using Power method**

Input data for Matrix:

Enter the order of square matrix: 3

Enter details of the matrix:

Enter the number at position (0,0): 2

Enter the number at position (0,1): -1

Enter the number at position (0,2): 0

Enter the number at position (1,0): -1

Enter the number at position (1,1): 2

Enter the number at position (1,2): -1

Enter the number at position (2,0): 0

Enter the number at position (2,1): -1

Enter the number at position (2,2): 2

Enter details of the initial eigen vector:

Enter the number at position (0,0): 1

Enter the number at position (1,0): 0

Enter the number at position (2,0): 0

Enter the tolerable error: 0.00005

A:

2 -1 0

-1 2 -1

0 -1 2

Eigen Vector:

-0.718593

1

-0.695621

Eigen Value: 3.41422

Figure 12.1: Output of pass case

Input data for Matrix:

Enter the order of square matrix: 2

Enter details of the matrix:

Enter the number at position (0,0): 0

Enter the number at position (0,1): 0

Enter the number at position (1,0): 2

Enter the number at position (1,1): 0

Enter details of the initial eigen vector:

Enter the number at position (0,0): 1

Enter the number at position (1,0): 1

Enter the tolerable error: 0.5

A:

0 0

2 0

Eigen Vector:

-nan

-nan

Eigen Value: -nan

Figure 12.2: Output of fail case

**Experiment 13: Program for solving ordinary differential equation by Runge-Kutta Method.**

PS C:\Users\admin\Documents\Saksham Gupta\MSIT> ./a.exe

Differential equation: (y^2 - x^2)/(y^2 + x^2)

Enter x0: 0

Enter y0: 1

Enter h: 0.2

Enter n: 10

X1= 0.2 Y1= 1.196

X2= 0.4 Y2= 1.37527

X3= 0.6 Y3= 1.53311

X4= 0.8 Y4= 1.66914

X5= 1 Y5= 1.7839

X6= 1.2 Y6= 1.87805

X7= 1.4 Y7= 1.95211

X8= 1.6 Y8= 2.00642

X9= 1.8 Y9= 2.04118

X10= 2 Y10= 2.05647

Figure 13.1: Output of program