## Outputs:

1. **<u>Bisection Method</u>**

```
PS D:\Projects\Bishnu\BIT-Patan\BIT-3rd-Semester\BIT203 NM\Notes\Practicals> cd "d:\Projects\Bishnu\BIT-Patan\BIT-3rd-Semester\BIT203 NM\Notes\Practicals\Lab1NonLinearEquation\" ; if ($?) { gcc BisectionMethod.c -o BisectionMethod } ; if ($?) { .\BisectionMethod }
9da14e57532c
 Enter 2 initial guesses:
0 1
Enter tolerable error:
0.01

 Step           x0              x1              x2              f(x2)
1               0.000000        1.000000        0.500000        0.053222
2               0.500000        1.000000        0.750000        -0.856061
3               0.500000        0.750000        0.625000        -0.356691
4               0.500000        0.625000        0.562500        -0.141294
5               0.500000        0.562500        0.531250        -0.041512
6               0.500000        0.531250        0.515625        0.006475

 Root is: 0.515625
```

2. **Newton Raphson Method:**

```
PS D:\Projects\Bishnu\BIT-Patan\BIT-3rd-Semester\BIT203 NM\Notes\Practicals\Lab5OrdinaryDifferen
ishnu\BIT-Patan\BIT-3rd-Semester\BIT203 NM\Notes\Practicals\Lab1NonLinearEquation\" ; if ($?) {
wtonRaphsonMethod } ; if ($?) { .\NewtonRaphsonMethod }
1-4d93-9f7d-ed019d9a2b00
 Enter initial guess:
0
Enter tolerable error:
0.001
Enter maximum iteration:
10

 Step            x0              f(x0)           x1              f(x1)
1               0.000000        -2.000000       0.666667        0.000000
2               0.666667        0.214113        0.607493        0.214113
3               0.607493        0.001397        0.607102        0.001397

 Root is: 0.607102
PS D:\Projects\Bishnu\BIT-Patan\BIT-3rd-Semester\BIT203 NM\Notes\Practicals\Lab1NonLinearEquatio
```

3. **Secant Method**

```
PS D:\Projects\Bishnu\BIT-Patan\BIT-3rd-Semester\BIT203 NM\Notes\Practicals\Lab1NonLin
earEquation> cd "d:\Projects\Bishnu\BIT-Patan\BIT-3rd-Semester\BIT203 NM\Notes\Practic
als\Lab1NonLinearEquation\" ; if ($?) { gcc SecantMethod.c -o SecantMethod } ; if ($?)
 { .\SecantMethod }
32c
 Enter initial guesses:
1 2
Enter tolerable Error:
0.01
Enter maximum iteration:
5

 Step            x0       x1       x2       f(x2)
1                1.000000         2.000000         2.200000         1.248001
2                2.000000         2.200000         2.088968         -0.062124
3                2.200000         2.088968         2.094233         -0.003554

 Root is: 2.094233
```

4. **Lagrange Interpolation:**

```
PS D:\Projects\Bishnu\BIT-Patan\BIT-3rd-Semester\BIT203 NM\Notes\Practicals\Lab1NonLin
earEquation> cd "d:\Projects\Bishnu\BIT-Patan\BIT-3rd-Semester\BIT203 NM\Notes\Practic
als\Lab2InterpolationAndApprox\" ; if ($?) { gcc LagrangeInterpolation.c -o LagrangeIn
terpolation } ; if ($?) { .\LagrangeInterpolation }
ee14624-b33c-4ccf-9f51-9da14e57532cEnter number of data: 5
Enter data:
x[1]=1
y[1]=2
x[2]=3
y[2]=4
x[3]=5
y[3]=6
x[4]=7
y[4]=8
x[5]=9
y[5]=2
Enter interpolation point:9
Interpolated value at 9.000 is 2.000
```

5. **Newton Interpolation using Forward Method**

```
PS D:\Projects\Bishnu\BIT-Patan\BIT-3rd-Semester\BIT203 NM\Notes\Practicals\Lab2Interp
olationAndApprox> cd "d:\Projects\Bishnu\BIT-Patan\BIT-3rd-Semester\BIT203 NM\Notes\Pr
acticals\Lab2InterpolationAndApprox\" ; if ($?) { gcc NewtonForwardInterpolation.c -o
NewtonForwardInterpolation } ; if ($?) { .\NewtonForwardInterpolation }terpolation };e
ee14624-b33c-4ccf-9f51-9da14e57532c
Enter number of data:
5
Enter2data:
y[0]=3
x[1]=4
y[1]=5
x[2]=6
y[2]=5
x[3]=4
y[3]=3
x[4]=4
y[4]=2

 Forward Difference Table

2.00    3.00    2.00    -2.00   0.00    3.00
4.00    5.00    0.00    -2.00   3.00
6.00    5.00    -2.00   1.00
4.00    3.00    -1.00
4.00    2.00
```

6. **<u>Newton Interpolation using Backward Method:</u>**

```
PS D:\Projects\Bishnu\BIT-Patan\BIT-3rd-Semester\BIT203 NM\Notes\Practicals\Lab2Interp
olationAndApprox> cd "d:\Projects\Bishnu\BIT-Patan\BIT-3rd-Semester\BIT203 NM\Notes\Pr
acticals\Lab2InterpolationAndApprox\" ; if ($?) { gcc NewtonBackwardInterpolation.c -o
 NewtonBackwardInterpolation } ; if ($?) { .\NewtonBackwardInterpolation }
dInterpolation };eee14624-b33c-4ccf-9f51-9da14e57532cEnter number of data:
5
Enter data:
x[0]=1
y[0]=2
x[1]=3
y[1]=5
x[2]=4
y[2]=6
x[3]=1
y[3]=2
x[4]=3
y[4]=4

 Backward Difference Table

1.00    2.00
3.00    5.00    3.00
4.00    6.00    1.00    -2.00
1.00    2.00    -4.00   -5.00   -3.00
3.00    4.00    2.00    6.00    11.00   14.00
```

7. **Newton Interpolation using Divided Method:**

```
PS D:\Projects\Bishnu\BIT-Patan\BIT-3rd-Semester\BIT203 NM\Notes\Practicals\Lab2Interp
olationAndApprox> cd "d:\Projects\Bishnu\BIT-Patan\BIT-3rd-Semester\BIT203 NM\Notes\Pr
acticals\Lab2InterpolationAndApprox\" ; if ($?) { gcc NewtonDividedInterpolation.c -o
NewtonDividedInterpolation } ; if ($?) { .\NewtonDividedInterpolation }
terpolation };eee14624-b33c-4ccf-9f51-9da14e57532c
 Enter the number of observations:
5

 Enter the different values of x:
1
2
3
4
5

 The corresponding values of y are:
2
4
6
8
10

 Enter the value of 'k' in f(k) you want to evaluate:
9

f(9)=18
```

8. **<u>Trapezoidal Rule:</u>**

```
PS D:\Projects\Bishnu\BIT-Patan\BIT-3rd-Semester\BIT203 NM\Notes\Practicals\Lab2Interp
olationAndApprox> cd "d:\Projects\Bishnu\BIT-Patan\BIT-3rd-Semester\BIT203 NM\Notes\Pr
acticals\Lab3NumericalDifferentiationAndIntegration\" ; if ($?) { gcc TrapezoidalRule.
c -o TrapezoidalRule } ; if ($?) { .\TrapezoidalRule }
14624-b33c-4ccf-9f51-9da14e57532cEnter lower limit of integration: 3
Enter upper limit of integration: 1
Enter number of sub intervals: 5

 Required value of integration is: -0.470
```

9. **<u>Simpson's 1/3 Rule:</u>**

```
PS D:\Projects\Bishnu\BIT-Patan\BIT-3rd-Semester\BIT203 NM\Notes\Practicals\Lab3NumericalDiff
Projects\Bishnu\BIT-Patan\BIT-3rd-Semester\BIT203 NM\Notes\Practicals\Lab3NumericalDifferenti
gcc Simpsons_1by3_Rule.c -o Simpsons_1by3_Rule } ; if ($?) { .\Simpsons_1by3_Rule }
Enter lower limit of integration: 2
Enter upper limit of integration: 1
Enter number of sub intervals: 4

 Required value of integration is: -0.322
```

10. **Simpson’s 3/8 Rule:**

```
PS D:\Projects\Bishnu\BIT-Patan\BIT-3rd-Semester\BIT203 NM\Notes\Practicals\Lab3NumericalDifferentiationA
-Patan\BIT-3rd-Semester\BIT203 NM\Notes\Practicals\Lab3NumericalDifferentiationAndIntegration\" ; if ($?)
by8_Rule } ; if ($?) { .\Simpsons_3by8_Rule }
Enter lower limit of integration: 2
Enter upper limit of integration: 0
Enter number of sub intervals: 5

 Required value of integration is: -1.062
```

## 11. **Gauss Elimination Method**

```
PS D:\Projects\Bishnu\BIT-Patan\BIT-3rd-Semester\BIT203 NM\Notes\Practicals> cd "d:\Pr
BIT203 NM\Notes\Practicals\Lab4LinearEquation\" ; if ($?) { gcc GaussElimination.c -o
mination }
ed019d9a2b00Enter number of unknowns:2
a[1][1]=1
a[1][2]=2
a[1][3]=3
a[2][1]=4
a[2][2]=6
a[2][3]=5

 Solution:
x[1]=-4.000
x[2]=3.500
```

12.**Gauss Jordan Method**

```
PS D:\Projects\Bishnu\BIT-Patan\BIT-3rd-Semester\BIT203 NM\Notes\Practicals\Lab4LinearEqua
n\BIT-3rd-Semester\BIT203 NM\Notes\Practicals\Lab4LinearEquation\" ; if ($?) { gcc GaussJo
\GaussJordan }
Enter coefficients of Augmented Matrix:
a[1][1]=1
a[1][2]=2
a[2][1]=4
a[2][2]=5
a[2][3]=6

 Solution:
x[1]=-1.000
x[2]=2.000
```

13.**Matrix Inversing Using Gauss Jordan Method:**

```
PS D:\Projects\Bishnu\BIT-Patan\BIT-3rd-Semester\BIT203 NM\Notes\Practicals\Lab4Linea
n\BIT-3rd-Semester\BIT203 NM\Notes\Practicals\Lab4LinearEquation\" ; if ($?) { gcc Ma
ionGaussJordan } ; if ($?) { .\MatrixInversionGaussJordan }
Enter number of unknowns:2
a[1][1]=1
a[1][2]=2
a[1][3]=3
a[2][1]=4
a[2][2]=6
a[2][3]=5

 Solution:
x[1]=-4.000
x[2]=3.500
```

14.**Matrix Factorization using Do Little LU Decomposition:**

```
PS D:\Projects\Bishnu\BIT-Patan\BIT-3rd-Semester\BIT203 NM\Notes\Practicals\Lab4LinearEquation
n\BIT-3rd-Semester\BIT203 NM\Notes\Practicals\Lab4LinearEquation\" ; if ($?) { gcc MatrixFacto
 -o MatrixFactorizationDoLittleLUDecomposition } ; if ($?) { .\MatrixFactorizationDoLittleLUDe
Enter the order of square matrix: 2

Enter matrix element:
Enter A[0][0] element: 1
Enter A[0][1] element: 2
Enter A[1][0] element: 3
Enter A[1][1] element: 4

Enter the constant terms:
B[0]1
B[1]2
[L]:
    1.000    0.000
    3.000    1.000


[U]:
    1.000    2.000
    0.000    4.000


[Y]:
    1.000   -1.000

[X]:
    1.500   -0.250
```

15.**<u>Matrix Factorization using Cholesky's Method</u>**

```
PS D:\Projects\Bishnu\BIT-Patan\BIT-3rd-Semester\BIT203 NM\Notes\Practicals\Lab4LinearEquatio
n\BIT-3rd-Semester\BIT203 NM\Notes\Practicals\Lab4LinearEquation\" ; if ($?) { gcc CholeskysM
($?) { .\CholeskysMethod }
5.00000 0.00000 0.00000
3.00000 3.00000 0.00000
-1.00000 1.00000 3.00000

4.24264 0.00000 0.00000 0.00000
5.18545 6.56591 0.00000 0.00000
12.72792 3.04604 1.64974 0.00000
9.89949 1.62455 1.84971 1.39262
PS D:\Projects\Bishnu\BIT-Patan\BIT-3rd-Semester\BIT203 NM\Notes\Practicals\Lab4LinearEquatio
                                                        Ln 16, Col 40   Spaces: 4   UTF-8
```

16.**Jacob Iterative Method:**

```
PS D:\Projects\Bishnu\BIT-Patan\BIT-3rd-Semester\BIT203 NM\Notes\Practicals\Lab4LinearEqu
n\BIT-3rd-Semester\BIT203 NM\Notes\Practicals\Lab4LinearEquation\" ; if ($?) { gcc JacobI
od } ; if ($?) { .\JacobIterativeMethod }
1-4d93-9f7d-ed019d9a2b00Enter tolerable error:
0.001

 Count   x       y       z
1        0.8500  -0.9000         1.2500
2        1.0200  -0.9650         1.0300
3        1.0013  -1.0015         1.0033
4        1.0004  -1.0000         0.9997

 Solution: x= 1.000, y = -1.000 and z = 1.000
```

17. **<u>Gauss Seidal Iterative Method:</u>**

```
PS D:\Projects\Bishnu\BIT-Patan\BIT-3rd-Semester\BIT203 NM\Notes\Practicals\Lab4LinearEquatior
n\BIT-3rd-Semester\BIT203 NM\Notes\Practicals\Lab4LinearEquation\" ; if ($?) { gcc GaussSeidal
\GaussSeidal }
Enter tolerable error:
0.001

Count   x       y       z
1       0.8500  -1.0275 1.0109
2       1.0025  -0.9998 0.9998
3       1.0000  -1.0000 1.0000

Solution: x=1.000, y=-1.000 and z = 1.000
```

18. **Power Method:**

```
PS D:\Projects\Bishnu\BIT-Patan\BIT-3rd-Semester\BIT203 NM\Notes\Practicals\Lab4LinearEq
n\BIT-3rd-Semester\BIT203 NM\Notes\Practicals\Lab4LinearEquation\" ; if ($?) { gcc Power
\PowerMethod }
Enter Order of Matrix:1
Enter Tolerable Error:2
Enter coefficient of Matrix:
a[1][1]=3
Enter Initial Guess Vector:
x[1]=4


STEP-1:
Eigen Value = 12.000000
Eigen Vector:
1.000000

STEP-2:
Eigen Value = 3.000000
Eigen Vector:
1.000000

STEP-3:
Eigen Value = 3.000000
Eigen Vector:
1.000000
```

19.**Taylor Series:**

```
PS D:\Projects\Bishnu\BIT-Patan\BIT-3rd-Semester\BIT203 NM\Notes\Practicals\Lab4LinearEquati
n\BIT-3rd-Semester\BIT203 NM\Notes\Practicals\Lab5OrdinaryDifferentialEquation\" ; if ($?) {
 } ; if ($?) { .\TaylorSeries }


 Enter the value of x in the series: 2

 Enter the number of terms in the series: 3

 The sum of the taylor series is: 5.00

PS D:\Projects\Bishnu\BIT-Patan\BIT-3rd-Semester\BIT203 NM\Notes\Practicals\Lab5OrdinaryDiff
```

20.**Picard's Method:**

```
PS D:\Projects\Bishnu\BIT-Patan\BIT-3rd-Semester\BIT203 NM\Notes\Practicals\Lab5OrdinaryDi
ishnu\BIT-Patan\BIT-3rd-Semester\BIT203 NM\Notes\Practicals\Lab5OrdinaryDifferentialEquati
-o PicardsMethod } ; if ($?) { .\PicardsMethod }

X
0.0000 0.4000 0.8000 1.2000 1.6000 2.0000 2.4000 2.8000

Y(1)
1.0000 1.4800 2.1200 2.9200 3.8800 5.0000 6.2800 7.7200

Y(2)
1.0000 1.5045 2.3419 3.7552 6.0645 9.6667 15.0352 22.7205

Y(3)
1.0000 1.5053 2.3692 3.9833 7.1131 13.1333 24.3249 44.2335
PS D:\Projects\Bishnu\BIT-Patan\BIT-3rd-Semester\BIT203 NM\Notes\Practicals\Lab5OrdinaryDi
```

## 21.**Euler’s Method:**

```
PS D:\Projects\Bishnu\BIT-Patan\BIT-3rd-Semester\BIT203 NM\Notes\Practicals\Lab5OrdinaryDifferentialEq
ishnu\BIT-Patan\BIT-3rd-Semester\BIT203 NM\Notes\Practicals\Lab5OrdinaryDifferentialEquation\" ; if ($
o EulersMethod } ; if ($?) { .\EulersMethod }
d-ed019d9a2b00Enter Initial Condition
x0 = 1
y0 = 2
Enter calculation point xn = 3
Enter number of steps: 4

x0      y0      slope   yn
------------------------------
1.0000  2.0000  3.0000  3.5000
1.5000  3.5000  5.0000  6.0000
2.0000  6.0000  8.0000  10.0000
2.5000  10.0000 12.5000 16.2500

Value of y at x = 3.00 is 16.250
```

22.**<u>Heun's Method</u>**

```
PS D:\Projects\Bishnu\BIT-Patan\BIT-3rd-Semester\BIT203 NM\Notes\Practicals\Lab5OrdinaryDifferent
ishnu\BIT-Patan\BIT-3rd-Semester\BIT203 NM\Notes\Practicals\Lab5OrdinaryDifferentialEquation\" ;
 HeunsMethod } ; if ($?) { .\HeunsMethod }
d019d9a2b00Program for Solution of Ordinary Differential Equation
Heun's Method
Enter value for x and y
1 2
Enter value for h and last of x
1 2
y = 8.500000    x = 2.000000
```

23.**<u>Range Kutta's Method</u>**

```
PS D:\Projects\Bishnu\BIT-Patan\BIT-3rd-Semester\BIT203 NM\Notes\Practicals\Lab5Ordi
ishnu\BIT-Patan\BIT-3rd-Semester\BIT203 NM\Notes\Practicals\Lab5OrdinaryDifferential
.c -o RangeKuttaMethod } ; if ($?) { .\RangeKuttaMethod }
Enter Initial Condition
x0 = 1
y0 = 2
Enter calculation point xn = 0.4
Enter number of steps: 4

x0      y0      yn
1.0000  2.0000  1.9049
0.8500  1.9049  1.7996
0.7000  1.7996  1.6837
0.5500  1.6837  1.5575

Value of y at x = 0.40 is 1.557
```

24.**Boundary Value Problem**

```
 Enter x0,y0,xn,yn,h:1 2 3 4 5

 Enter the trial M1:1

6.000000        164.291672
B1 is 164.291672
Enter the value of M2:2

6.000000        190.125000
B2 is 190.125000
M2=2.000000     M1=1.000000
Exact value of M =-5.204840
6.000000        -1.#IND00
PS D:\Projects\Bishnu\BIT-Patan\BIT-3rd-Semester\BIT203 NM\Notes\Practicals\Lab5OrdinaryDifferentialEquation>
```

25.**Shooting Method:**

```
  Enter x0,y0,xn,yn,h:1 2 3 4 5

  Enter the trial M1:1

6.000000        164.291672
B1 is 164.291672
Enter the value of M2:2

6.000000        190.125000
B2 is 190.125000
M2=2.000000     M1=1.000000
Exact value of M =-5.204840
6.000000        -1.#IND00
PS D:\Projects\Bishnu\BIT-Patan\BIT-3rd-Semester\BIT203 NM\Notes\Practicals\Lab5OrdinaryDifferent
```