Course Code	**:**	**BCSL-058**
Course Title	**:**	**Computer oriented Numerical techniques Lab**
Assignment Number	**:**	**BCA(V)/L-058/Assignment/2026-27**
Maximum Marks	**:**	**50**
Weightage	**:**	**25%**
Last Dates for Submission	**:**	**31st October, 2026 (For July 2026 Session)**
	:	**30th April, 2027 (For January 2027 Session)**

This assignment has eight problems of 40 marks, each of 5 marks. All problems are compulsory. 10 marks are for viva voce. Please go through the guidelines regarding assignments given in the programme guide for the format of presentation.

Q1: Write a Program to find the solution of a system of linear equations using Gaussian Elimination Methods: **(5 Marks)**

$$X + Y + Z = 2$$
$$X - 2Y + 3Z = 14$$
$$X + 3Y - 6Z = -23$$

Q2: Write a program to implement the Gauss-Seidel method for finding the roots of linear equations. **(5 Marks)**

Q3: Write a program to implement the Bisection method for finding a positive root of the equation $X^2 - 9x + 21 = 0$. You have to make a suitable choice for the bounds. **(5 Marks)**

Q4: Write a program in C for the demonstration of Bessel's Formula for Interpolation. **(5 Marks)**

Q5: Write a program in C for the demonstration of Lagrange's Interpolation method for finding f(x) for a given input value of x. **(5 Marks)**

Q6: Write a program in C to approximate first and second derivatives of $y = f(x)$ at $x = x_0$ using Newton's Forward Difference Formula (FD). **(5 Marks)**

Q7: Write a program in C to find the approximate value of the following definite integral using the Trapezoidal formula: **(5 Marks)**

$$\int_0^{\Pi/4} tanx\, dx$$

Q8: Write a program in C to demonstrate Newton's Divided Difference method. **(5 Marks)**

Q1: Write a Program to find the solution of a system of linear equations using Gaussian Elimination Methods: (5 Marks)

X + Y +Z = 2

X-2Y+3Z = 14

X + 3Y – 6Z = -23

Answer

Gaussian elimination converts the three given equations into an upper-triangular system. We then use back substitution to calculate the unknowns. Partial pivoting is included in the C program to avoid dividing by a zero or very small pivot.

Subtract the first equation from the second and third: −3Y + 2Z = 12 and 2Y − 7Z = −25. Eliminating Y gives −17Z/3 = −17, so Z = 3. Next, Y = −2 and X = 1. Substitution in all three original equations gives 2, 14 and −23 respectively.

C program

```
#include <stdio.h>
#include <math.h>
#define N 3
int main(void) {
    double a[N][N+1] = {
        {1, 1, 1, 2},
        {1,-2, 3,14},
        {1, 3,-6,-23}
    };
    double x[N];
    for (int k=0; k<N-1; k++) {
        int pivot=k;
        for (int i=k+1; i<N; i++)
            if (fabs(a[i][k])>fabs(a[pivot][k])) pivot=i;
        if (fabs(a[pivot][k])<1e-12) {
            puts("No unique solution (zero pivot)."); return 1;
        }
        if (pivot!=k)
            for (int j=0; j<=N; j++) {
                double temp=a[k][j];
                a[k][j]=a[pivot][j];a[pivot][j]=temp;
            }
        for (int i=k+1; i<N; i++) {
            double factor=a[i][k]/a[k][k];
            for (int j=k; j<=N; j++)
                a[i][j]-=factor*a[k][j];
        }
    }
    if (fabs(a[N-1][N-1])<1e-12) {
        puts("No unique solution.");return 1;
    }
    for (int i=N-1; i>=0; i--) {
        double sum=a[i][N];
        for (int j=i+1; j<N; j++) sum-=a[i][j]*x[j];
```