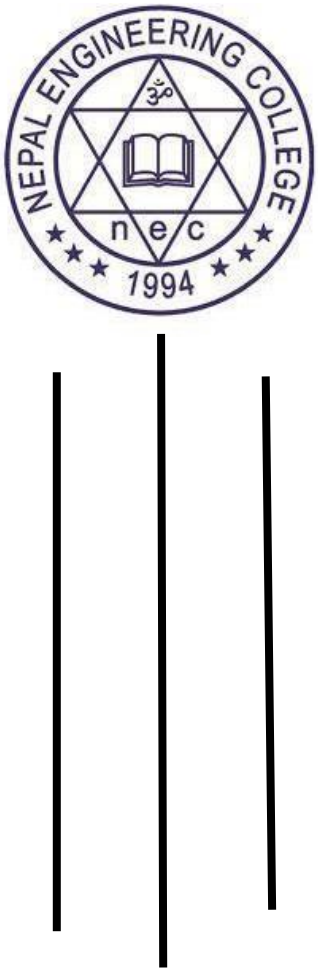
NEPAL ENGINEERING COLLEGE

(AFFILIATED TO POKHARA UNIVERSITY)

Changunarayan, Bhaktapur



REPORT ON:

Basic Gauss Elimination Method

SUBMITTED BY:

NAME: Subash Khanal

CRN: 020-626

SUBMITTED TO:

Electrical and

Electronics

Experiment no:-6

TITLE:-

Gauss Elimination Method

OBJECTIVE:-

To find the solution of number of equation using Matlab and C-programming.

THEORY:-

Introduction:

Gauss elimination method proposes a systematic strategy for reducing the system of equations to the upper triangular form using the forward elimination approach and then for obtaining values of unknowns using the back substitution process. For general set of 3 equations:

$$a_{11}x + a_{12}y + a_{13}z = b_1$$

$$a_{21}x + a_{22}y + a_{23}z = b_2$$

$$a_{31}x + a_{32}y + a_{33}z = b_3$$

The final form of elimination process is given by:

$$\begin{matrix} a_{11} & a_{12} & a_{13} & = b_1 \\ 0 & a_{22} & a_{23} & = b_2 \\ 0 & 0 & a_{33} & = b_3 \end{matrix}$$

Value of x, y and z can be computed as:

$$z = \frac{b_3}{a_{33}},$$

$$y = \frac{b_2 - a_{23}\, z}{a_{22}},$$

$$x = \frac{b_1 - a_{13}\, z - a_{12} y}{a_{11}},$$

Algorithm:

1. Start

2. Read Number of Unknowns: n

3. Read Augmented Matrix (A) of n by n+1 Size

4. Transform Augmented Matrix (A) to Upper Triangular Matrix by Row Operations.

5. Obtain Solution by Back Substitution.

6. Display Result.

7. Stop

Question:-

Write a MATLAB program to solve following system of linear equations using Gauss elimination method:

$$x + y + z = 6$$

$$2x - y + 3z = 4$$

$$4x + 5y - 10z = 13$$

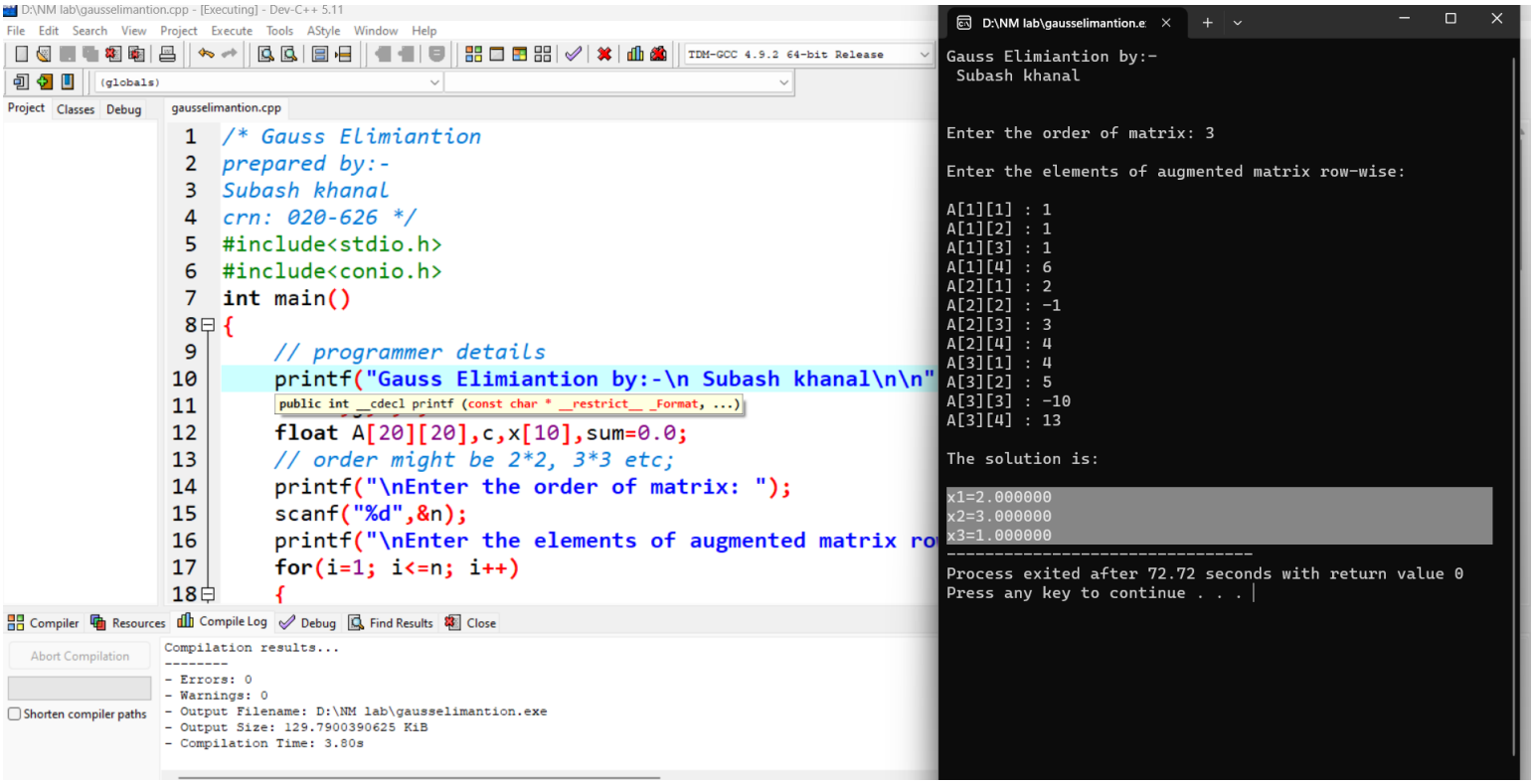
Using C-programming

Syntax:-

```
/* Gauss Elimiantion
prepared by:-
Subash khanal
crn: 020-626 */
#include<stdio.h>
#include<conio.h>
int main()
{
    // programmer details
    printf("Gauss Elimiantion by:-\n Subash khanal\n\n");
    int i,j,k,n;
    float A[20][20],c,x[10],sum=0.0;
    // order might be 2*2, 3*3 etc;
    printf("\nEnter the order of matrix: ");
    scanf("%d",&n);
    printf("\nEnter the elements of augmented matrix row-wise:\n\n");
    for(i=1; i<=n; i++)
    {
        for(j=1; j<=(n+1); j++)
        {
            printf("A[%d][%d] : ", i,j);
            scanf("%f",&A[i][j]);
        }
    }
    /* loop for the generation of upper triangular matrix*/
    for(j=1; j<=n; j++)
    {
        for(i=1; i<=n; i++)
        {
            if(i>j)
            {
                c=A[i][j]/A[j][j];
```

```
                for(k=1; k<=n+1; k++)
                {
                    A[i][k]=A[i][k]-c*A[j][k];
                }
            }
        }
    }
    x[n]=A[n][n+1]/A[n][n];
    /* this loop is for backward substitution*/
    for(i=n-1; i>=1; i--)
    {
        sum=0;
        for(j=i+1; j<=n; j++)
        {
            sum=sum+A[i][j]*x[j];
        }
        x[i]=(A[i][n+1]-sum)/A[i][i];
    }
    printf("\nThe solution is: \n");
    for(i=1; i<=n; i++)
    {
        //printing the value of solution
        printf("\nx%d=%f\t",i,x[i]);
    }
    return(0);
}
```

Output:-



The value of x_1 x_2 x_3 are found by using C-programming i.e. 2, 3, 1 respectively which was highlighted on output screen.

Using Matlab.

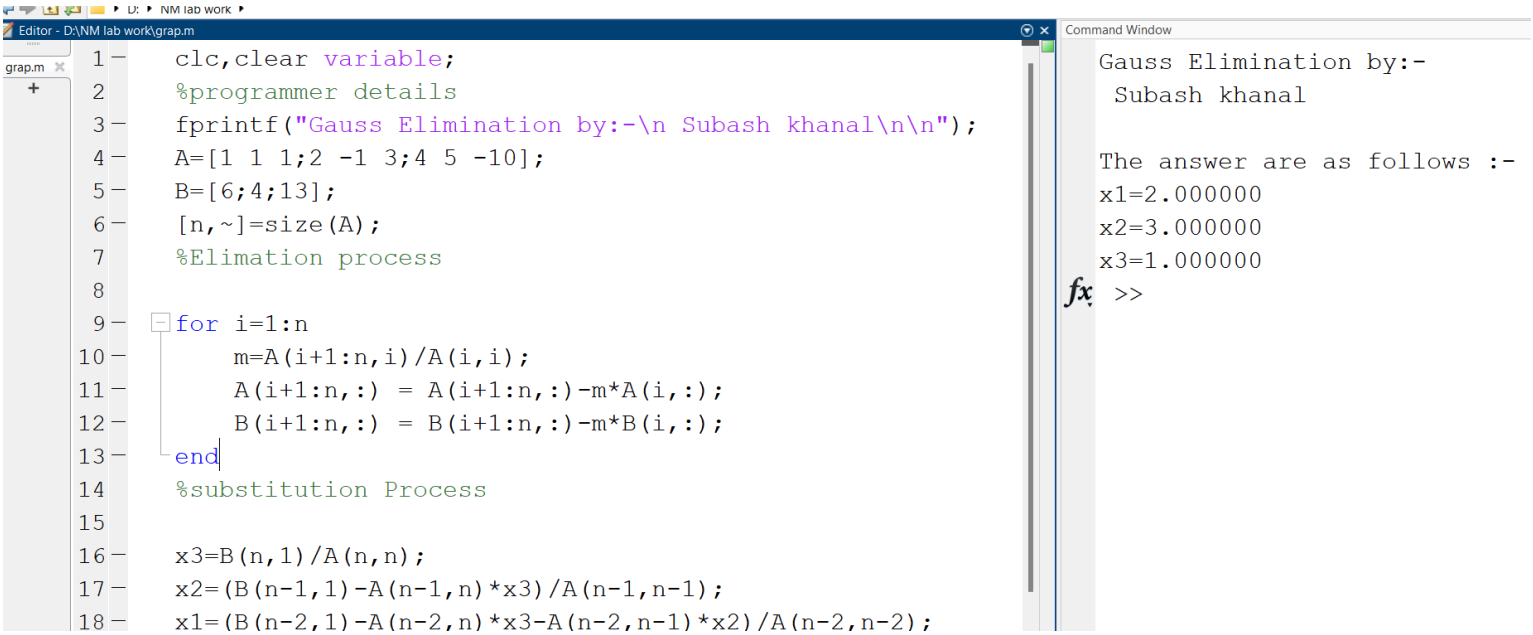
Syntax:-

```
clc,clear variable;
%programmer details
fprintf("Gauss Elimination by:-\n Subash
khanal\n\n");
A=[1 1 1;2 -1 3;4 5 -10];
B=[6;4;13];
[n,~]=size(A);
%Elimation process

for i=1:n
    m=A(i+1:n,i)/A(i,i);
    A(i+1:n,:) = A(i+1:n,:)-m*A(i,:);
    B(i+1:n,:) = B(i+1:n,:)-m*B(i,:);
end
%substitution Process

x3=B(n,1)/A(n,n);
x2=(B(n-1,1)-A(n-1,n)*x3)/A(n-1,n-1);
x1=(B(n-2,1)-A(n-2,n)*x3-A(n-2,n-1)*x2)/A(n-
2,n-2);
disp('The answer are as follows :-');
fprintf('x1=%f\nx2=%f\nx3=%f\n ',x1,x2,x3);
```

Output:-



Using Matlab the value of x_1 x_2 x_3 are found to 2, 3, 1 respectively.

Description:-

From above program of c-programming and Matlab it was clear that value are same using Gauss Elimination method. So, using above program we can find the x_1 x_2 x_3 are found to 2, 3, 1 respectively.

Conclusion:-

Hence, from above we can implement and calculate the value using the Gauss Elimination on Matlab and C-programming.