NEPAL ENGINEERING COLLEGE

(AFFILIATED TO POKHARA UNIVERSITY)

Changunarayan, Bhaktapur

REPORT ON:

Gauss Jacobi Method

SUBMITTED BY:

NAME: Subash Khanal

CRN: 020-626

SUBMITTED TO:

Electrical and

Electronics

Experiment no:-7

TITLE:-

Gauss Jacobi Method

OBJECTIVE:-

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

THEORY:-

Introduction:

Gauss Jacobi method is an iterative algorithm for determining the solutions of a system of linear equations. It follows the principle of direct substitution where the values of unknowns are improved by substituting directly the previous values. Diagonally dominant matrix is sufficient criterion for the convergence of Jacobi method. Let us consider a system of 3 equations with 3 unknowns:

$$a_{11}x_1 + a_{12}x_2 + a_{13}x_3 = b_1$$

$$a_{21}x_1 + a_{22}x_2 + a_{23}x_3 = b_2$$

$$a_{31}x_1 + a_{32}x_2 + a_{33}x_3 = b_3$$

We rewrite the original system as,

$$x_1 = \frac{a_{12}x_2 + a_{13}x_3 - b_1}{-a_{11}}$$
$$x_2 = \frac{a_{21}x_1 + a_{23}x_3 - b_2}{-a_{22}}$$
$$x_3 = \frac{a_{31}x_1 + a_{32}x_2 - b_3}{-a_{33}}$$

In general, it can be written as,

$$x_i = (-\frac{1}{a_{ii}}) \sum_{j=1}^{n} a(i,j) * x(j) - b_i \qquad [for\ i \neq j]$$

We can compute the $x1$, $x2 and\ x3$ by using initial guesses for these values. These new values are again used to compute the next set of x values. The process can continue till we obtain a desired level of accuracy in the x values.

Algorithm:

```
1. Read matrix A, matrix B, max_itr and error limit
2. Set initial guesses and x_new values to (0,0,0)
3. For iteration = 1,2,3………. Max_itr
    i) For i =1, 2, 3,………..n
    ii) Set sum = 0
    iii) For j = 1, 2, 3……..n ($j \neq i$)
        $sum$ = $sum$ + $a(i,j) * x(j)$
        $Repeat\ j$
    iv) Set $x_{new}(i)$ = (- $\frac{1}{a_{ii}}$)(sum-B(i))
        $Repeat\ i$
    v) if $abs(xnew - x\ )$ < $error\ limit$:
        go to step 4
    vi) $x$ = $xnew$
     $Repeat\ iteration$
    4. Display $x$ values
```

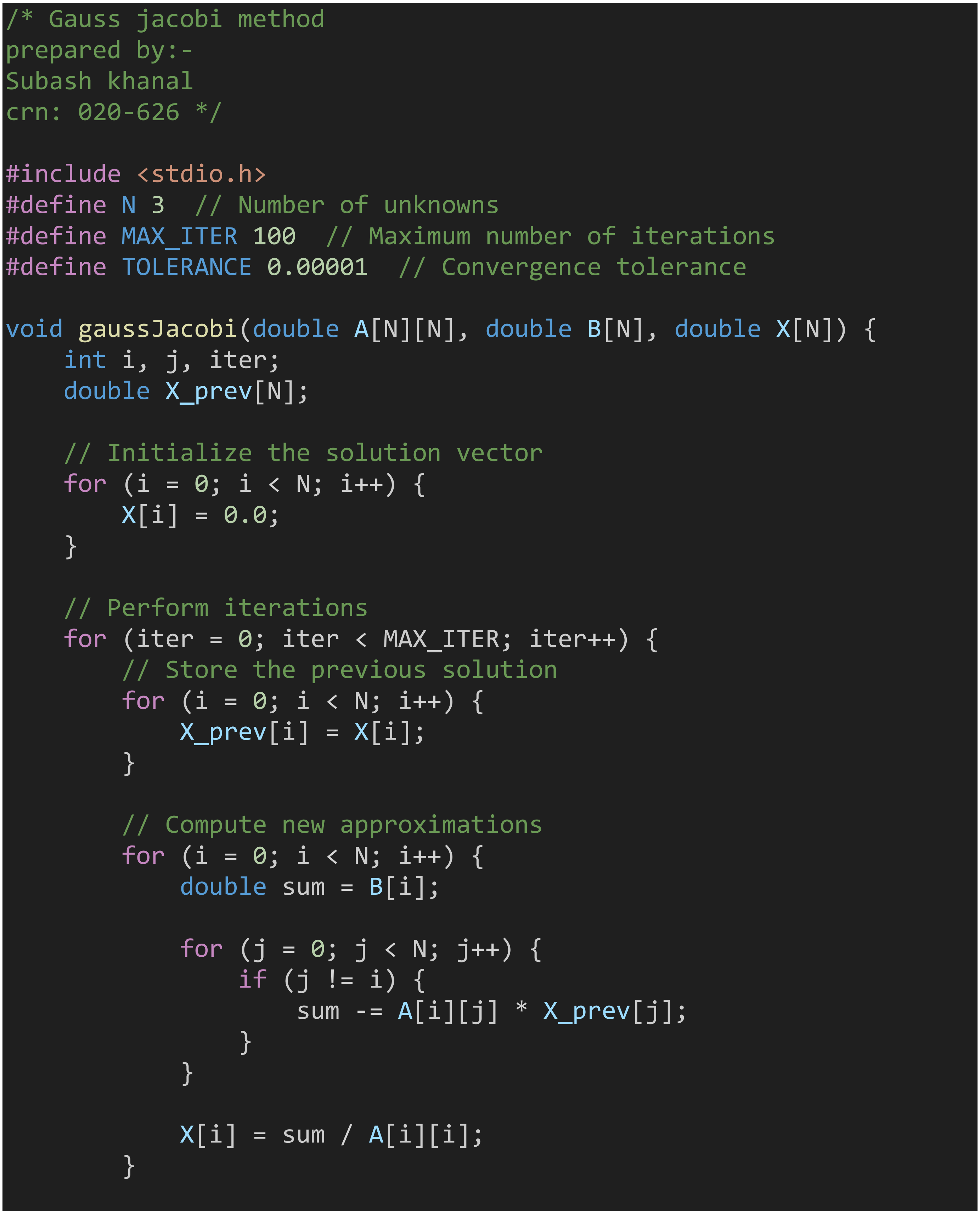
Question:-

Implement above algorithm to solve the given set of linear equations:

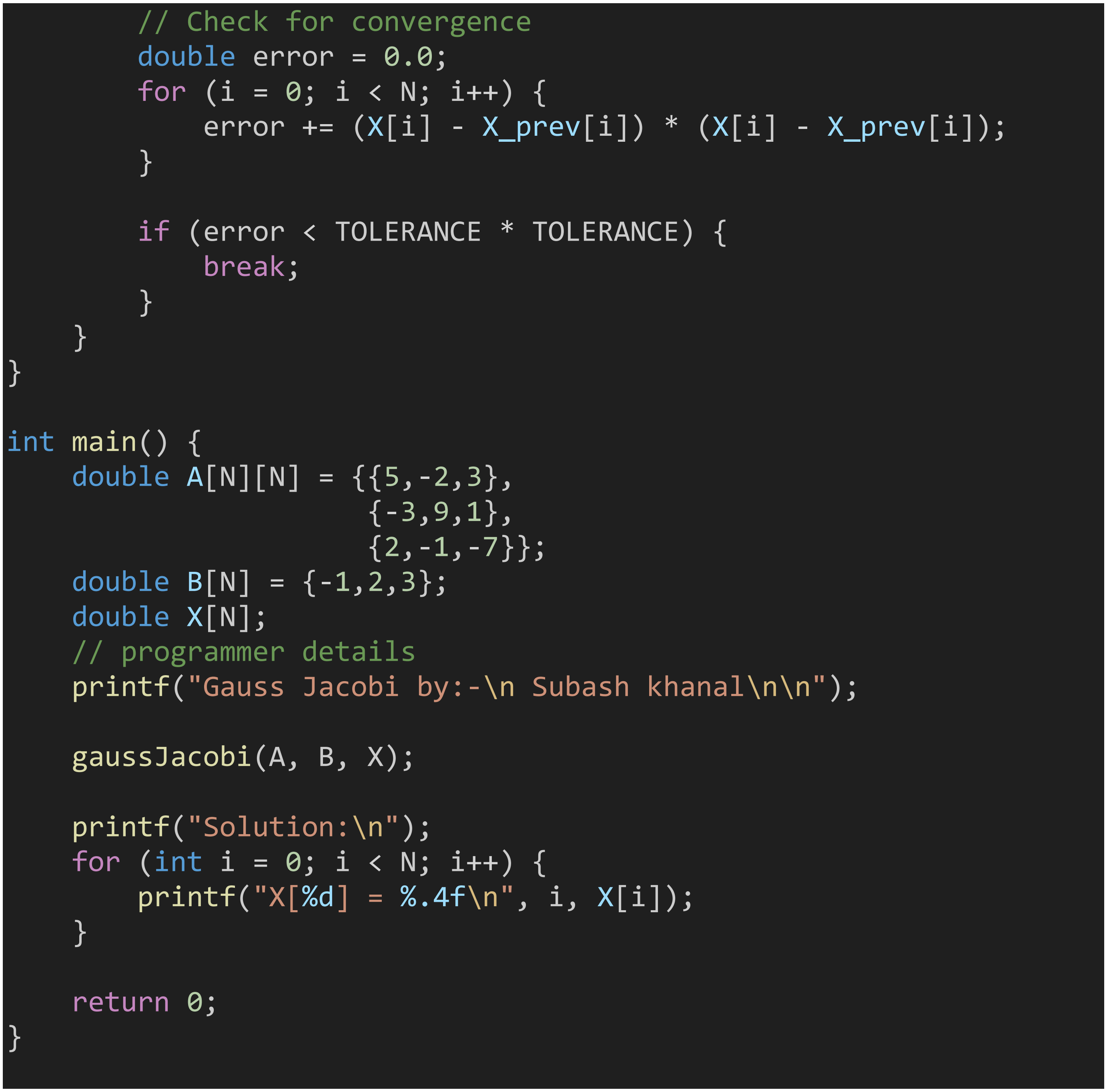
$$5x_1 - 2x_2 + 3x_3 = -1$$
$$-3x_1 + 9x_2 + x_3 = 2$$
$$2x_1 - x_2 - 7x_3 = 3$$

Using C-programming

Syntax:-

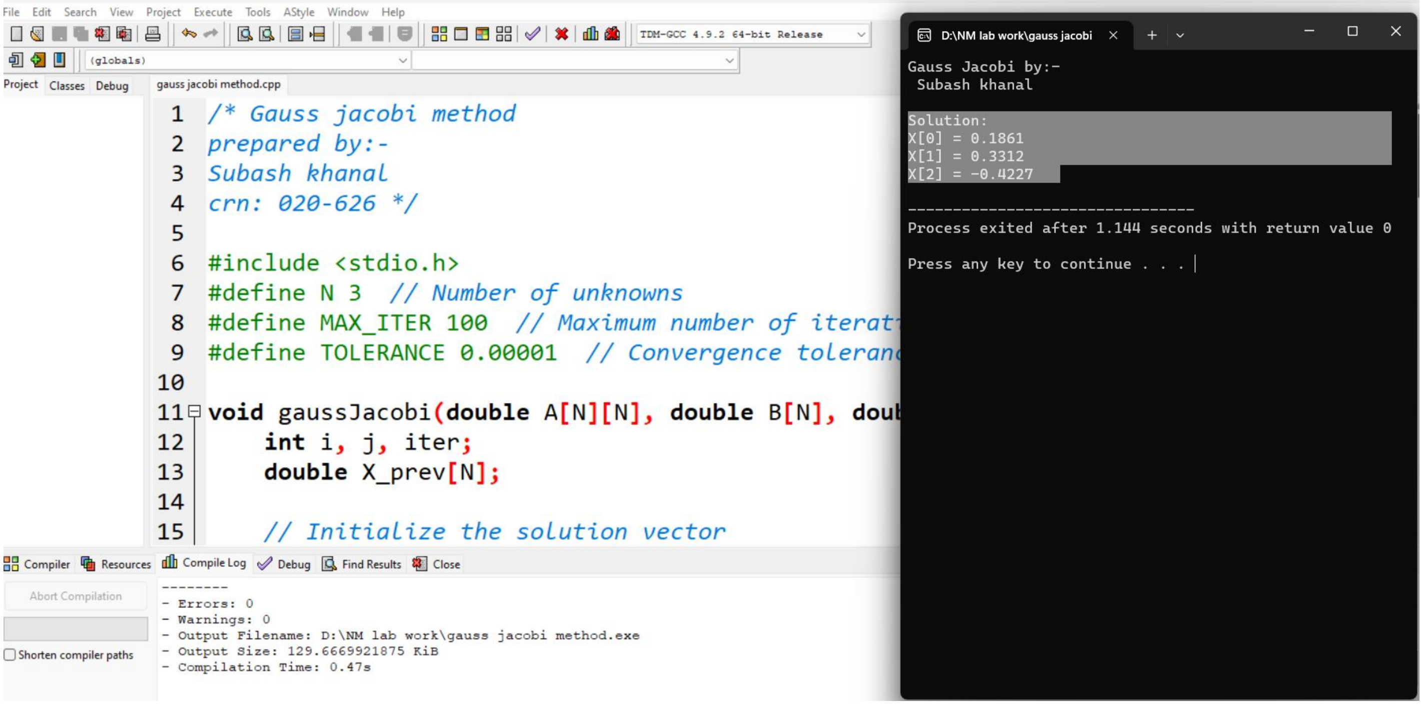


```c
/* Gauss jacobi method
prepared by:-
Subash khanal
crn: 020-626 */

#include <stdio.h>
#define N 3  // Number of unknowns
#define MAX_ITER 100  // Maximum number of iterations
#define TOLERANCE 0.00001  // Convergence tolerance

void gaussJacobi(double A[N][N], double B[N], double X[N]) {
    int i, j, iter;
    double X_prev[N];

    // Initialize the solution vector
    for (i = 0; i < N; i++) {
        X[i] = 0.0;
    }

    // Perform iterations
    for (iter = 0; iter < MAX_ITER; iter++) {
        // Store the previous solution
        for (i = 0; i < N; i++) {
            X_prev[i] = X[i];
        }

        // Compute new approximations
        for (i = 0; i < N; i++) {
            double sum = B[i];

            for (j = 0; j < N; j++) {
                if (j != i) {
                    sum -= A[i][j] * X_prev[j];
                }
            }

            X[i] = sum / A[i][i];
        }
```



```c
        // Check for convergence
        double error = 0.0;
        for (i = 0; i < N; i++) {
            error += (X[i] - X_prev[i]) * (X[i] - X_prev[i]);
        }

        if (error < TOLERANCE * TOLERANCE) {
            break;
        }
    }
}

int main() {
    double A[N][N] = {{5,-2,3},
                      {-3,9,1},
                      {2,-1,-7}};
    double B[N] = {-1,2,3};
    double X[N];
    // programmer details
    printf("Gauss Jacobi by:-\n Subash khanal\n\n");

    gaussJacobi(A, B, X);

    printf("Solution:\n");
    for (int i = 0; i < N; i++) {
        printf("X[%d] = %.4f\n", i, X[i]);
    }

    return 0;
}
```

Output:-


File Edit Search View Project Execute Tools AStyle Window Help
TDM-GCC 4.9.2 64-bit Release
(globals)
Project Classes Debug
gauss jacobi method.cpp
Compiler Resources Compile Log Debug Find Results Close
Abort Compilation
□ Shorten compiler paths
D:\NM lab work\gauss jacobi

```
/* Gauss jacobi method
prepared by:-
Subash khanal
crn: 020-626 */

#include <stdio.h>
#define N 3  // Number of unknowns
#define MAX_ITER 100  // Maximum number of iterat
#define TOLERANCE 0.00001  // Convergence toleran

void gaussJacobi(double A[N][N], double B[N], dou
    int i, j, iter;
    double X_prev[N];

    // Initialize the solution vector
```

```
--------
- Errors: 0
- Warnings: 0
- Output Filename: D:\NM lab work\gauss jacobi method.exe
- Output Size: 129.6669921875 KiB
- Compilation Time: 0.47s
```

```
Gauss Jacobi by:-
 Subash khanal

Solution:
X[0] = 0.1861
X[1] = 0.3312
X[2] = -0.4227

--------------------------------
Process exited after 1.144 seconds with return value 0

Press any key to continue . . .
```

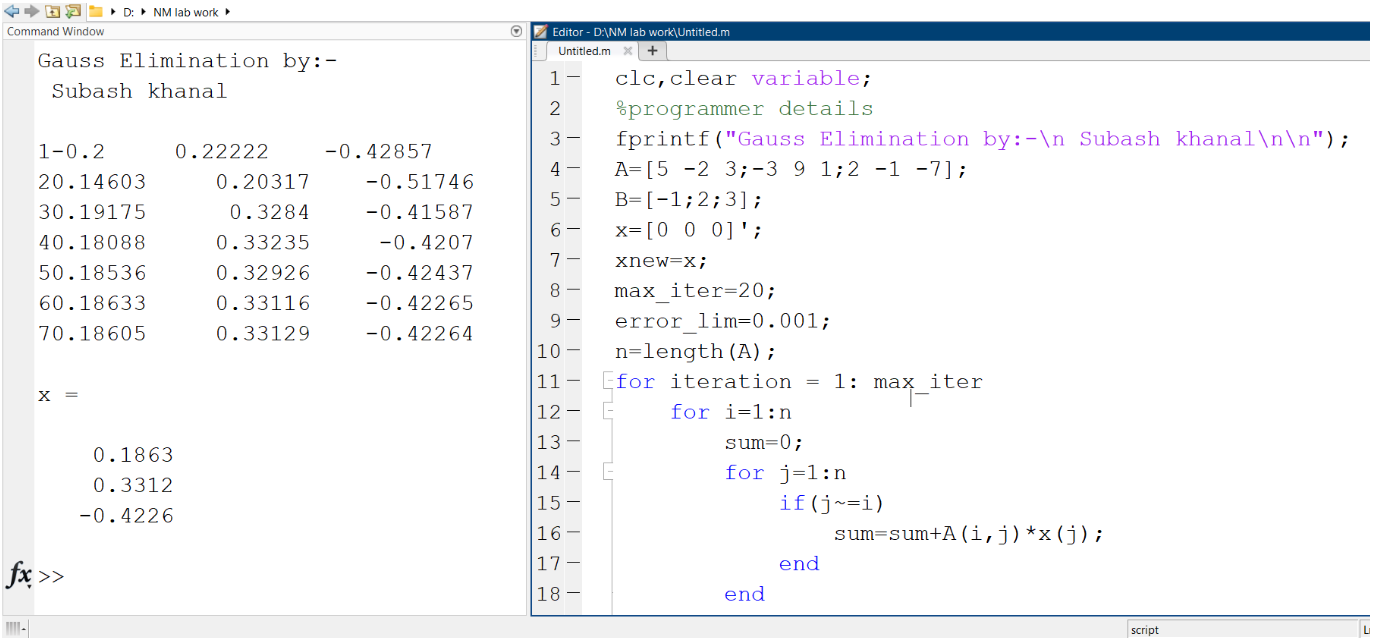
The value of x1 x2 x3 are found by using C-programming i.e. 0.1861, 0.3312, -0.4227respectively which was highlighted on output screen.

Using Matlab.

Syntax:-

```
clc,clear variable;
%programmer details
fprintf("Gauss Elimination by:-\n Subash
khanal\n\n");
A=[5 -2 3;-3 9 1;2 -1 -7];
B=[-1;2;3];
x=[0 0 0]';
xnew=x;
max_iter=20;
error_lim=0.001;
n=length(A);
for iteration = 1: max_iter
    for i=1:n
        sum=0;
        for j=1:n
            if(j~=i)
                sum=sum+A(i,j)*x(j);
            end
        end
        xnew(i)=(-1/A(i,i))*(sum-B(i));
    end
    disp([num2str(iteration)
num2str(xnew')])
    err=abs(xnew-x);
    if(err<error_lim)
        break;
    end
    x=xnew;
end
x % display the value of x1,x2,x3.
```

Output:-



```
Gauss Elimination by:-
 Subash khanal

1-0.2       0.22222     -0.42857
20.14603      0.20317     -0.51746
30.19175       0.3284     -0.41587
40.18088      0.33235      -0.4207
50.18536      0.32926     -0.42437
60.18633      0.33116     -0.42265
70.18605      0.33129     -0.42264


x =

    0.1863
    0.3312
   -0.4226

fx >>
```

```
clc,clear variable;
%programmer details
fprintf("Gauss Elimination by:-\n Subash khanal\n\n");
A=[5 -2 3;-3 9 1;2 -1 -7];
B=[-1;2;3];
x=[0 0 0]';
xnew=x;
max_iter=20;
error_lim=0.001;
n=length(A);
for iteration = 1: max_iter
    for i=1:n
        sum=0;
        for j=1:n
            if(j~=i)
                sum=sum+A(i,j)*x(j);
            end
        end
```

Using Matlab the value of $x1$ $x2$ $x3$ are found to 0.1861, 0.3312, -0.4227 respectively.

Description:-

From above program of c-programming and Matlab it was clear that value are same using Gauss Jacobi method. So, using above program we can find the $x1$,$x2$ $x3$ are 0.1861, 0.3312, -0.4227 found to respectively.

Conclusion:-

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