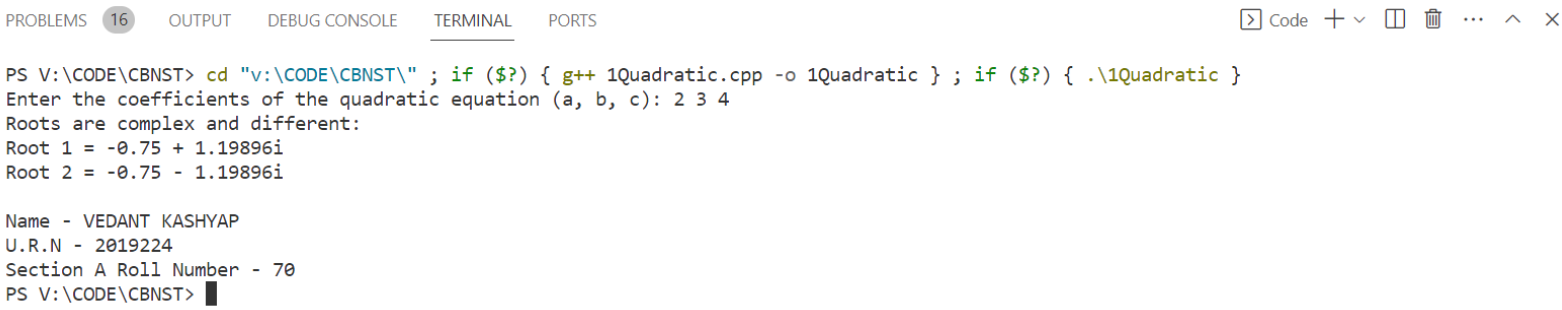
```
/*
Name – VEDANT KASHYAP
U.R.N - 2019224
Class Roll Number - 70
Method Name- Quadratic Equation
*/
#include <bits/stdc++.h>
using namespace std;
int main()
{

    double a, b, c;
    cout << "Enter the coefficients of the quadratic equation (a, b, c): ";
    cin >> a >> b >> c;

    double discriminant = b * b - 4 * a * c;

    if (discriminant > 0)
    {
        double root1 = (-b + sqrt(discriminant)) / (2 * a);
        double root2 = (-b - sqrt(discriminant)) / (2 * a);
        cout << "Roots are real and different:\n";
        cout << "Root 1 = " << root1 << "\n";
        cout << "Root 2 = " << root2 << "\n";
    }
    else if (discriminant == 0)
    {
        double root = -b / (2 * a);
        cout << "Roots are real and identical:\n";
        cout << "Root = " << root << "\n";
    }
    else
    {
        double realPart = -b / (2 * a);
        double imaginaryPart = sqrt(-discriminant) / (2 * a);
        cout << "Roots are complex and different:\n";
        cout << "Root 1 = " << realPart << " + " << imaginaryPart << "i\n";
        cout << "Root 2 = " << realPart << " - " << imaginaryPart << "i\n";
    }
    cout << endl;
    cout << "Name - VEDANT KASHYAP"
         << "\n"
         << "U.R.N - 2019224"
         << "\n"
         << "Section A Roll Number - 70";
    return 0;
}
```

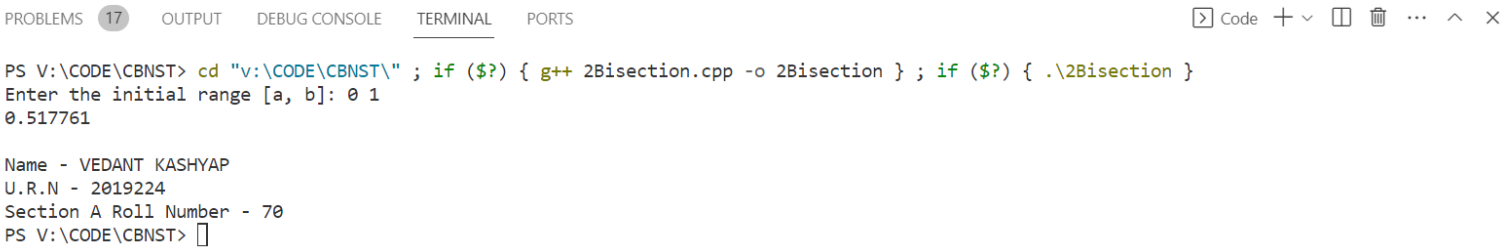
OUTPUT :


PROBLEMS 16
OUTPUT
DEBUG CONSOLE
TERMINAL
PORTS
Code
PS V:\CODE\CBNST> cd "v:\CODE\CBNST\" ; if ($?) { g++ 1Quadratic.cpp -o 1Quadratic } ; if ($?) { .\1Quadratic }
Enter the coefficients of the quadratic equation (a, b, c): 2 3 4
Roots are complex and different:
Root 1 = -0.75 + 1.19896i
Root 2 = -0.75 - 1.19896i
Name - VEDANT KASHYAP
U.R.N - 2019224
Section A Roll Number - 70
PS V:\CODE\CBNST>

```
/*
Name – VEDANT KASHYAP
U.R.N - 2019224
Class Roll Number - 70
Method Name - Bisection Method
*/
#include <bits/stdc++.h>
using namespace std;

constexpr double e = 2.71828182846;

double f(double x) {
    return (cos(x) - x * pow(e,x));
}

double calc(double a, double b) {
    if (f(a) * f(b) > 0) {
        cout << "Root is out of bound" << endl;
        return 0.0;
    } else if (f(a) == 0) {
        return a;
    } else if (f(b) == 0) {
        return b;
    }

    double c = a;
    while ((b - a) >= 0.0001) {
        c = (a + b) / 2.0;
        if (f(c) == 0.0) {
            break;
        } else if (f(c) * f(a) < 0) {
            b = c;
        } else {
            a = c;
        }
    }
    return c;
}

int main() {
    double a, b;
    cout << "Enter the initial range [a, b]: ";
    cin >> a >> b;
    double x = calc(a, b);
    cout << << x << endl;

    cout<<endl;
```

```
    cout << "Name - VEDANT KASHYAP" << "\n" << "U.R.N - 2019224" << "\n" <<
"Section A Roll Number - 70";

    return 0;


}
```

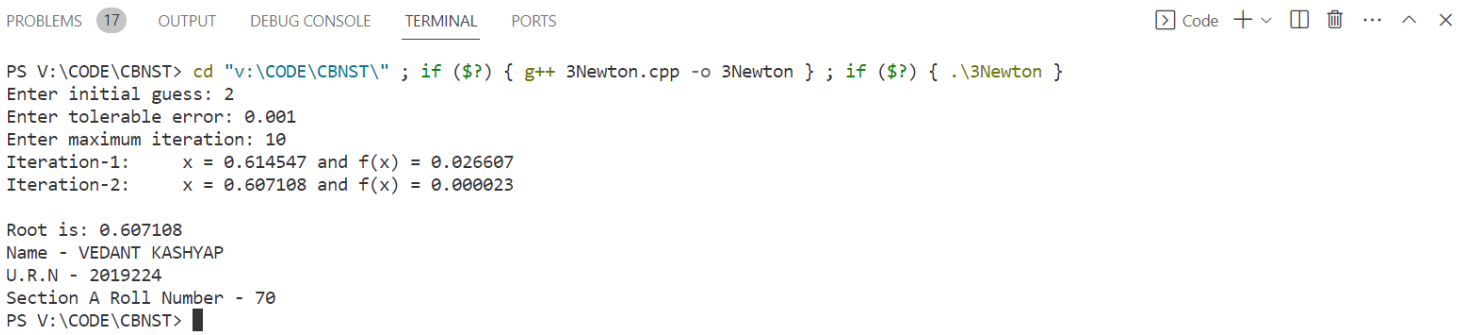
OUTPUT :


PROBLEMS 17
OUTPUT
DEBUG CONSOLE
TERMINAL
PORTS
Code
PS V:\CODE\CBNST> cd "v:\CODE\CBNST\" ; if ($?) { g++ 2Bisection.cpp -o 2Bisection } ; if ($?) { .\2Bisection }
Enter the initial range [a, b]: 0 1
0.517761
Name - VEDANT KASHYAP
U.R.N - 2019224
Section A Roll Number - 70
PS V:\CODE\CBNST>

```
/*
Name – VEDANT KASHYAP
U.R.N - 2019224
Class Roll Number - 70
Method Name- Newton Raphson
*/

#include <bits/stdc++.h>
using namespace std;


#define    f(x)    3*x - cos(x) -1

#define   g(x)   3 + sin(x)



int main()
{
    float x0, x1, f0, f1, g0, e;
    int step = 1, N;

    /* Setting precision and writing floating point values in fixed-point notation. */
    cout<< setprecision(6)<< fixed;

    /* Inputs */
    cout<<"Enter initial guess: ";
    cin>>x0;
    cout<<"Enter tolerable error: ";
    cin>>e;
    cout<<"Enter maximum iteration: ";
    cin>>N;

    do
    {
        g0 = g(x0);
        f0 = f(x0);
        if(g0 == 0.0)
        {
            cout<<"Mathematical Error.";
            exit(0);
        }


        x1 = x0 - f0/g0;


        cout<<"Iteration-"<< step<<":\t x = "<< x1<<" and f(x) = "<< f(x1)<< endl;
```

```
        x0 = x1;

        step = step+1;

        if(step > N)
        {
            cout<<"Not Convergent.";
            exit(0);
        }

        f1 = f(x1);

     }while(fabs(f1)>e);

    cout<< endl<<"Root is: "<< x1;
    cout<<endl;
    cout << "Name - VEDANT KASHYAP" << "\n" << "U.R.N - 2019224" << "\n" <<
"Section A Roll Number - 70";
    return 0;
}
```

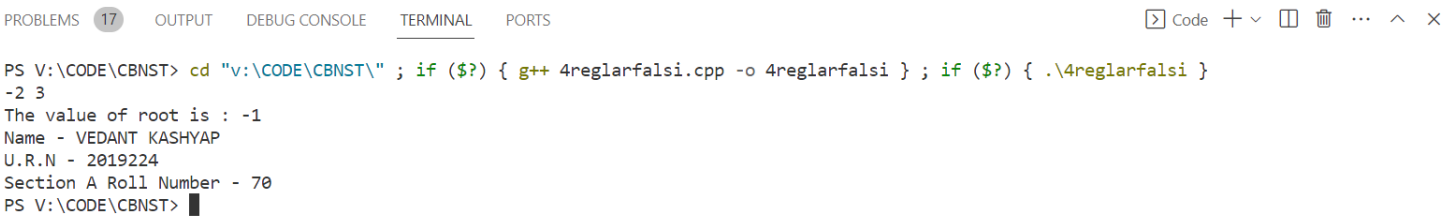
OUTPUT :


PROBLEMS 17
OUTPUT
DEBUG CONSOLE
TERMINAL
PORTS
Code
PS V:\CODE\CBNST> cd "v:\CODE\CBNST\" ; if ($?) { g++ 3Newton.cpp -o 3Newton } ; if ($?) { .\3Newton }
Enter initial guess: 2
Enter tolerable error: 0.001
Enter maximum iteration: 10
Iteration-1: x = 0.614547 and f(x) = 0.026607
Iteration-2: x = 0.607108 and f(x) = 0.000023
Root is: 0.607108
Name - VEDANT KASHYAP
U.R.N - 2019224
Section A Roll Number - 70
PS V:\CODE\CBNST>

```
/*
Name – VEDANT KASHYAP
U.R.N - 2019224
Class Roll Number - 70
Method Name- Regular Falsi
*/
#include <bits/stdc++.h>
using namespace std;
#define MAX_ITER 1000000

// Regular Falsi Method. The function is x^3 - x^2 + 2
double func(double x)
{
    return x * x * x - x * x + 2;
}
// Prints root of func(x) in interval [a, b]
void regulaFalsi(double a, double b)
{
    if (func(a) * func(b) >= 0)
    {
        cout << "You have not assumed right a and b\n";
        return;
    }
    double c = a;
    for (int i = 0; i < MAX_ITER; i++)
    {

        c = (a * func(b) - b * func(a)) / (func(b) - func(a));

        if (func(c) == 0)
            break;
        else if (func(c) * func(a) < 0)
            b = c;
        else
            a = c;
    }
    cout << "The value of root is : " << c;
}
// Driver program to test above function
int main()
{
    // Initial values assumed
    double a, b;
    cin >> a >> b;
    regulaFalsi(a, b);

    cout<<endl;
```

```
    cout << "Name - VEDANT KASHYAP" << "\n" << "U.R.N - 2019224" << "\n" <<
"Section A Roll Number - 70";
    return 0;


}
```

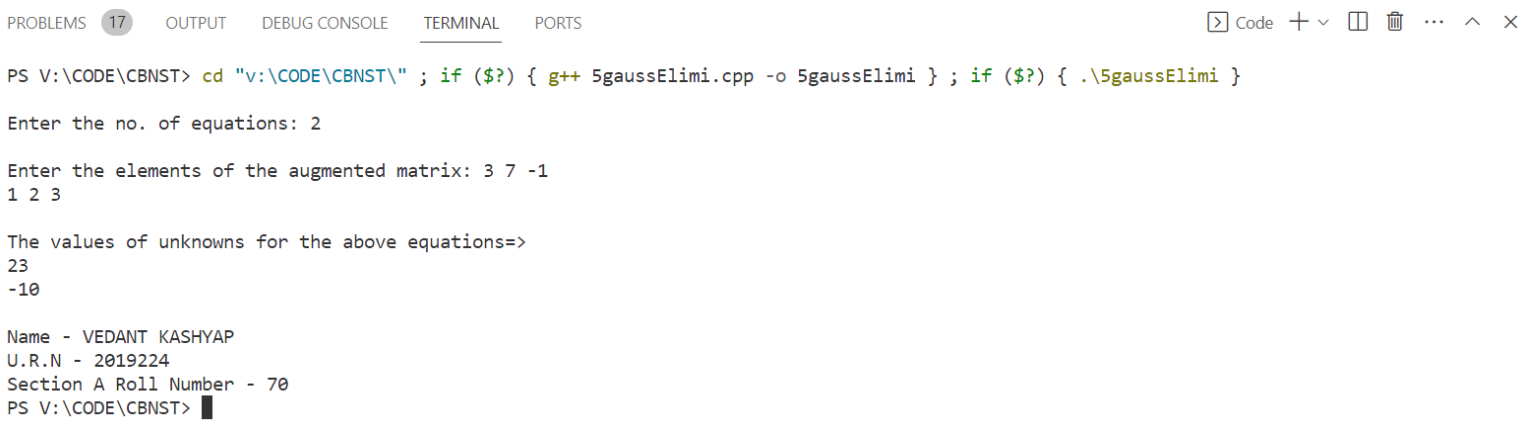
OUTPUT :


PROBLEMS 17
OUTPUT
DEBUG CONSOLE
TERMINAL
PORTS
Code
PS V:\CODE\CBNST> cd "v:\CODE\CBNST\" ; if ($?) { g++ 4reglarfalsi.cpp -o 4reglarfalsi } ; if ($?) { .\4reglarfalsi }
-2 3
The value of root is : -1
Name - VEDANT KASHYAP
U.R.N - 2019224
Section A Roll Number - 70
PS V:\CODE\CBNST>

```
/*
Name – VEDANT KASHYAP
U.R.N - 2019224
Class Roll Number - 70
Method Name- Gauss Elimination
*/
#include <bits/stdc++.h>
using namespace std;
int main()
{
    int i, j, k, n;
    cout << "\nEnter the no. of equations: ";
    cin >> n;

    float mat[n][n + 1];
    float res[n];
    cout << "\nEnter the elements of the augmented matrix: ";
    for (i = 0; i < n; i++)
    {
        for (j = 0; j < n + 1; j++)
        {
            cin >> mat[i][j];
        }
    }
    for (i = 0; i < n; i++)
    {
        for (j = i + 1; j < n; j++)
        {
            if (abs(mat[i][i]) < abs(mat[j][i]))
            {
                for (k = 0; k < n + 1; k++)
                {
                    /* swapping mat[i][k] and mat[j][k] */
                    mat[i][k] = mat[i][k] + mat[j][k];
                    mat[j][k] = mat[i][k] - mat[j][k];
                    mat[i][k] = mat[i][k] - mat[j][k];
                }
            }
        }
    }
    /* performing Gaussian elimination */
    for (i = 0; i < n - 1; i++)
    {
        for (j = i + 1; j < n; j++)
        {
            float f = mat[j][i] / mat[i][i];
            for (k = 0; k < n + 1; k++)
```

```
            {
                mat[j][k] = mat[j][k] - f * mat[i][k];
            }
        }
    }
    /* Backward substitution for discovering values of unknowns */
    for (i = n - 1; i >= 0; i--)
    {
        res[i] = mat[i][n];
        for (j = i + 1; j < n; j++)
        {
            if (i != j)
            {
                res[i] = res[i] - mat[i][j] * res[j];
            }
        }
        res[i] = res[i] / mat[i][i];
    }
    cout << "\nThe values of unknowns for the above equations=>\n";
    for (i = 0; i < n; i++)
    {
        cout << res[i] << "\n";
    }
    cout<<endl;
    cout << "Name - VEDANT KASHYAP" << "\n" << "U.R.N - 2019224" << "\n" <<
"Section A Roll Number - 70";
    return 0;
}
```

OUTPUT :


PROBLEMS 17
OUTPUT
DEBUG CONSOLE
TERMINAL
PORTS
Code
PS V:\CODE\CBNST> cd "v:\CODE\CBNST\" ; if ($?) { g++ 5gaussElimi.cpp -o 5gaussElimi } ; if ($?) { .\5gaussElimi }
Enter the no. of equations: 2
Enter the elements of the augmented matrix: 3 7 -1
1 2 3
The values of unknowns for the above equations=>
23
-10
Name - VEDANT KASHYAP
U.R.N - 2019224
Section A Roll Number - 70
PS V:\CODE\CBNST>

```
/*
Name – VEDANT KASHYAP
U.R.N - 2019224
Class Roll Number - 70
Method Name- Gauss Jordan
*/
#include <bits/stdc++.h>
#define SIZE 10
using namespace std;
int main()
{
   float a[SIZE][SIZE], x[SIZE], ratio;
   int i, j, k, n;

   cout << setprecision(3) << fixed;
   /* Inputs */
   /* 1. Reading number of unknowns */
   cout << "Enter number of unknowns: ";
   cin >> n;
   /* 2. Reading Augmented Matrix */
   cout << "Enter Coefficients of Augmented Matrix: " << endl;
   for (i = 1; i <= n; i++)
   {
      for (j = 1; j <= n + 1; j++)
      {
         cout << "a[" << i << "]" << j << "]= ";
         cin >> a[i][j];
      }
   }
   /* Applying Gauss Jordan Elimination */
   for (i = 1; i <= n; i++)
   {
      if (a[i][i] == 0.0)
      {
         cout << "Mathematical Error!";
         exit(0);
      }
      for (j = 1; j <= n; j++)
      {
         if (i != j)
         {
            ratio = a[j][i] / a[i][i];
            for (k = 1; k <= n + 1; k++)
            {
               a[j][k] = a[j][k] - ratio * a[i][k];
            }
         }
```

```
        }
    }
    /* Obtaining Solution */
    for (i = 1; i <= n; i++)
    {
        x[i] = a[i][n + 1] / a[i][i];
    }
    /* Displaying Solution */
    cout << endl
         << "Solution: " << endl;
    for (i = 1; i <= n; i++)
    {
        cout << "x[" << i << "] = " << x[i] << endl;
    }

    cout<<endl;
    cout << "Name - VEDANT KASHYAP" << "\n" << "U.R.N - 2019224" << "\n" <<
"Section A Roll Number - 70";

    return 0;
}
```

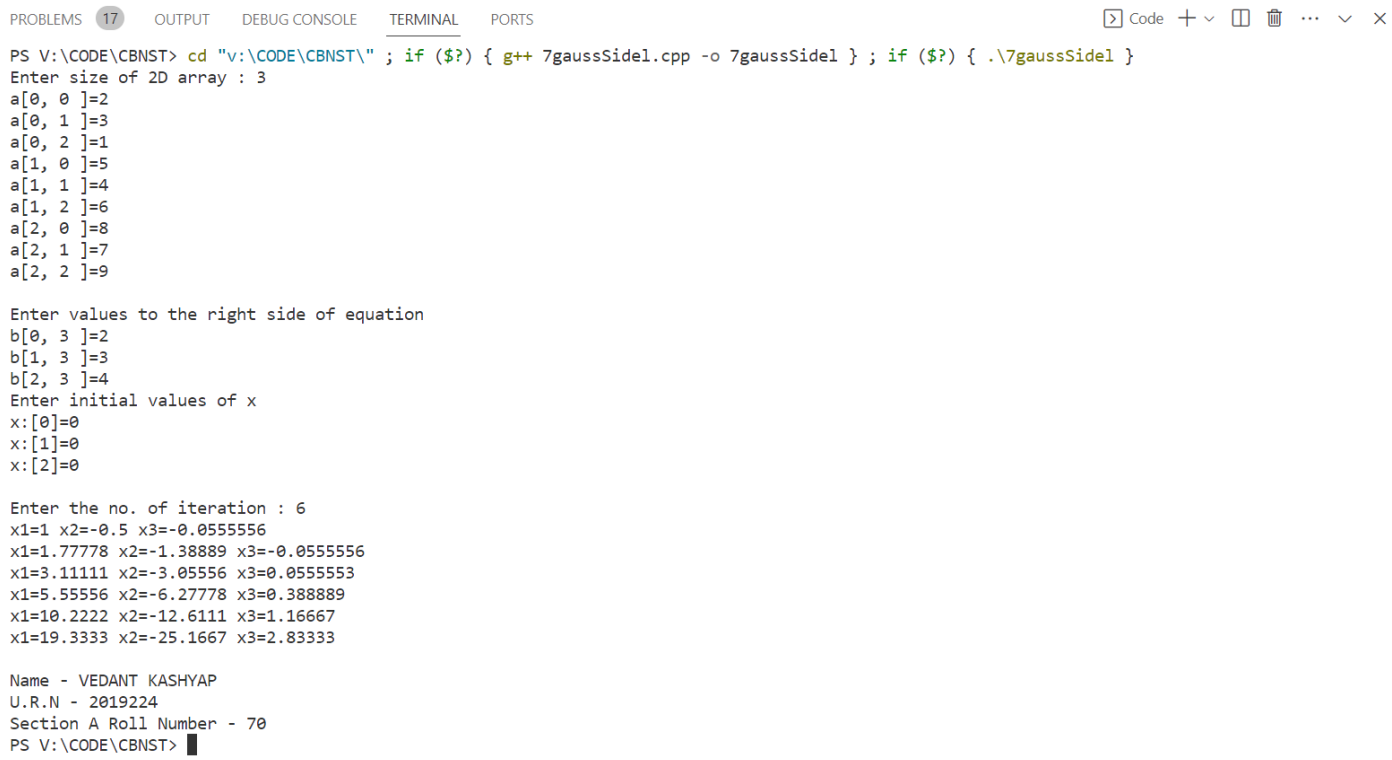
OUTPUT :


PROBLEMS 17
OUTPUT
DEBUG CONSOLE
TERMINAL
PORTS
Code
PS V:\CODE\CBNST> cd "v:\CODE\CBNST\" ; if ($?) { g++ 6gaussJordan.cpp -o 6gaussJordan } ; if ($?) { .\6gaussJordan }
Enter number of unknowns: 3
Enter Coefficients of Augmented Matrix:
a[1]1]= 2
a[1]2]= 3
a[1]3]= 5
a[1]4]= 6
a[2]1]= 1
a[2]2]= 8
a[2]3]= 6
a[2]4]= 2
a[3]1]= 8
a[3]2]= 9
a[3]3]= 1
a[3]4]= 2
Solution:
x[1] = 1.044
x[2] = -0.850
x[3] = 1.292
Name - VEDANT KASHYAP
U.R.N - 2019224
Section A Roll Number - 70
PS V:\CODE\CBNST>

```
/*
Name – VEDANT KASHYAP
U.R.N - 2019224
Class Roll Number - 70
Method Name- Gauss Seidel
*/
#include <bits/stdc++.h>

using namespace std;
int main(void)
{

    float a[10][10], b[10], m[10], n[10];
    int p = 0, q = 0, i = 0, j = 0;
    cout << "Enter size of 2D array : ";
    cin >> p;
    for (i = 0; i < p; i++)
    {
        for (j = 0; j < p; j++)
        {
            cout << "a[" << i << ", " << j << " ]=";
            cin >> a[i][j];
        }
    }
    cout << "\nEnter values to the right side of equation\n";
    for (i = 0; i < p; i++)
    {
        cout << "b[" << i << ", " << j << " ]=";
        cin >> b[i];
    }
    cout << "Enter initial values of x\n";
    for (i = 0; i < p; i++)
    {
        cout << "x:[" << i << "]=";
        cin >> m[i];
    }
    cout << "\nEnter the no. of iteration : ";
    cin >> q;
    while (q > 0)
    {
        for (i = 0; i < p; i++)
        {
            n[i] = (b[i] / a[i][i]);
            for (j = 0; j < p; j++)
            {
                if (j == i)
                    continue;
```

```
            n[i] = n[i] - ((a[i][j] / a[i][i]) * m[j]);
            m[i] = n[i];
        }
        cout << "x" << i + 1 << "=" << n[i] << " ";
    }
    cout << "\n";
    q--;
  }

  cout<<endl;
  cout << "Name - VEDANT KASHYAP" << "\n" << "U.R.N - 2019224" << "\n" <<
"Section A Roll Number - 70";
  return 0;
}
```

OUTPUT :


PROBLEMS 17
OUTPUT
DEBUG CONSOLE
TERMINAL
PORTS
Code
PS V:\CODE\CBNST> cd "v:\CODE\CBNST\" ; if ($?) { g++ 7gaussSidel.cpp -o 7gaussSidel } ; if ($?) { .\7gaussSidel }
Enter size of 2D array : 3
a[0, 0]=2
a[0, 1]=3
a[0, 2]=1
a[1, 0]=5
a[1, 1]=4
a[1, 2]=6
a[2, 0]=8
a[2, 1]=7
a[2, 2]=9
Enter values to the right side of equation
b[0, 3]=2
b[1, 3]=3
b[2, 3]=4
Enter initial values of x
x:[0]=0
x:[1]=0
x:[2]=0
Enter the no. of iteration : 6
x1=1 x2=-0.5 x3=-0.0555556
x1=1.77778 x2=-1.38889 x3=-0.0555556
x1=3.11111 x2=-3.05556 x3=0.0555553
x1=5.55556 x2=-6.27778 x3=0.388889
x1=10.2222 x2=-12.6111 x3=1.16667
x1=19.3333 x2=-25.1667 x3=2.83333
Name - VEDANT KASHYAP
U.R.N - 2019224
Section A Roll Number - 70
PS V:\CODE\CBNST>