MATHEMATICS FOR DATA SCIENCE

LAB ASSIGNMENT-2

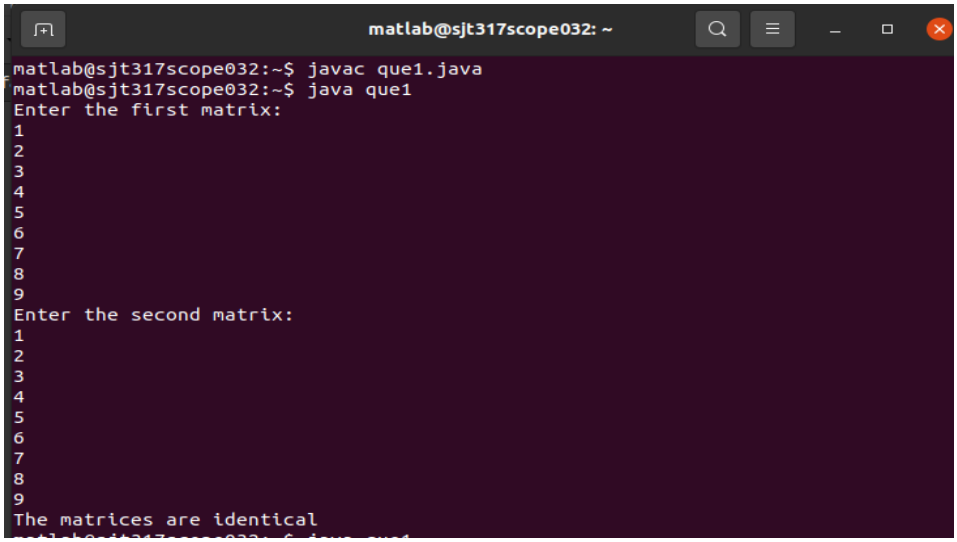
1. Write a program to check whether two given matrices are identical.

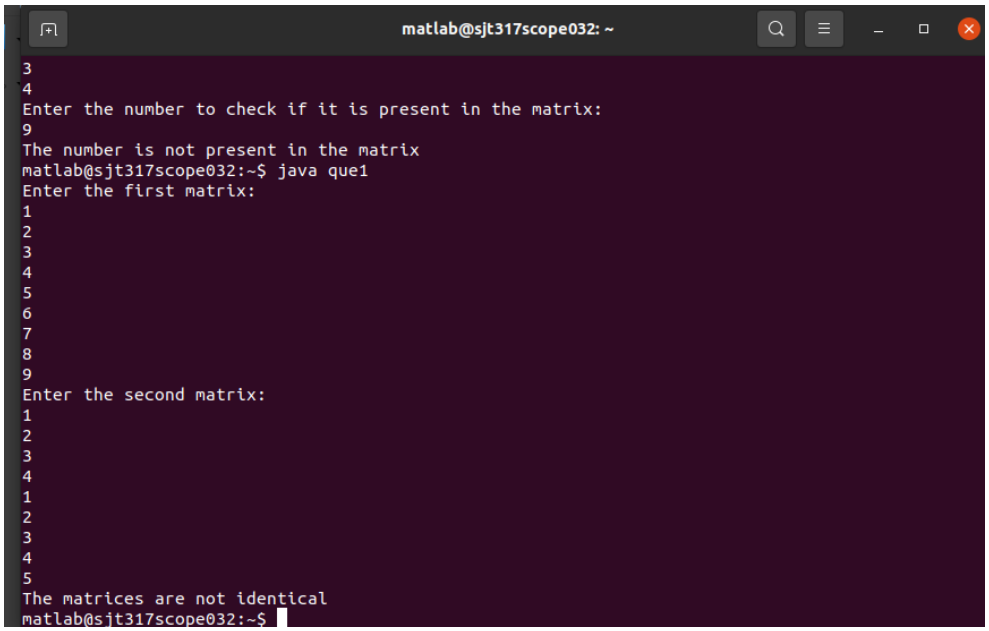
Code:

```
import java.lang.*;
import java.util.*;
class que1
{
public static void main(String args[])
 {
 int arr[][]=new int [10][10];
 Scanner sc=new Scanner(System.in);
 System.out.println("Enter the first matrix:");
 for(int i=0;i<3;i++)
 {for(int j=0;j<3;j++)
 {
 arr[i][j]=sc.nextInt();

 }}
 int arr1[][]=new int[10][10];
 System.out.println("Enter the second matrix:");
 for(int i=0;i<3;i++)
 {for(int j=0;j<3;j++)
 {
 arr1[i][j]=sc.nextInt();

 }}
 int flag=0;
 for(int i=0;i<3;i++)
 {
 for(int j=0;j<3;j++)
 {
 if(arr[i][j]!=arr1[i][j])
 {flag=1;
 break;
 }
 else
 {continue;}
 }

 }
 if(flag==1)
 {System.out.println("The matrices are not identical");}
```

```
else if(flag==0)
{System.out.println("The matrices are identical");}
}}
```

Output:



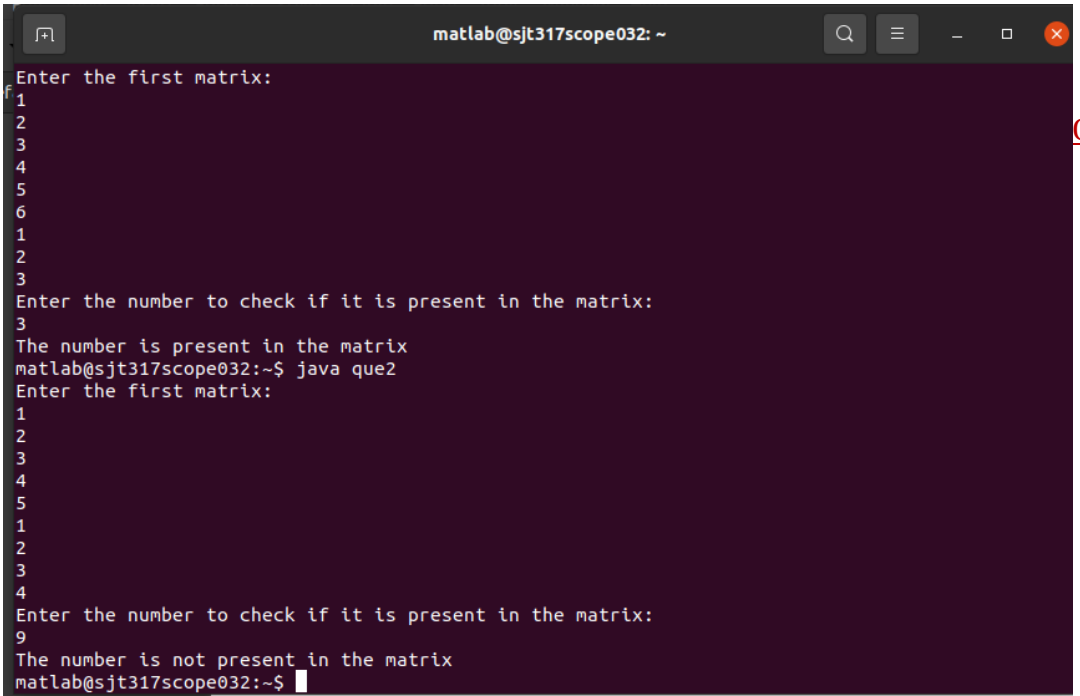


2. Write a program to check whether a given number is present in a matrix as one of the elements of the matrix.

Code:

```
import java.lang.*;
import java.util.*;

class que2
{public static void main(String args[])
{int arr[][]=new int [10][10];int num;
 Scanner sc=new Scanner(System.in);
 System.out.println("Enter the first matrix:");
 for(int i=0;i<3;i++)
 {for(int j=0;j<3;j++)
 {
 arr[i][j]=sc.nextInt();

 }}
 Scanner sc1=new Scanner(System.in);
 System.out.println("Enter the number to check if it is present in the matrix:");
 num=sc1.nextInt();
 int flag=0;
 for(int i=0;i<3;i++)
 {
 for(int j=0;j<3;j++)
 {
 if(arr[i][j]==num)
 {flag=1;
 break;
 }
 else
 {continue;}
 }

 }
 if(flag==1)
 {System.out.println("The number is present in the matrix");}
 else if(flag==0)
 {System.out.println("The number is not present in the matrix");}
 }}
```

Output:

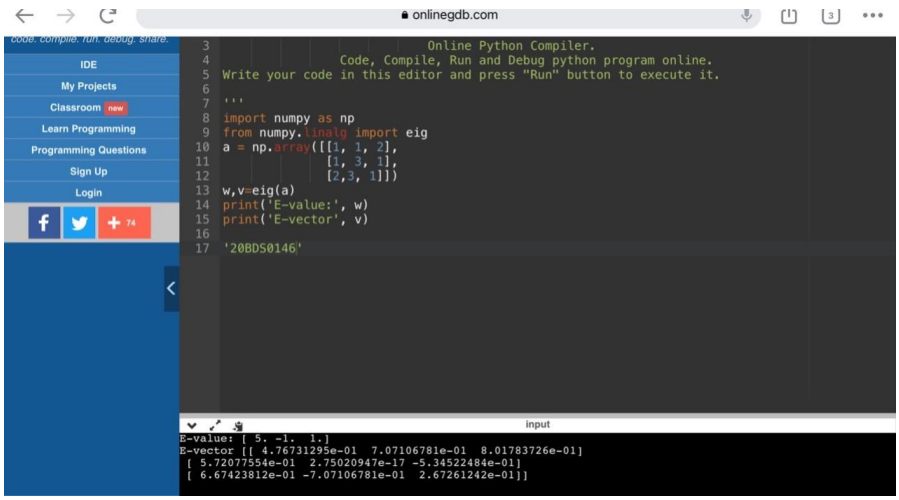


3. Write a program to find the Eigen values and Eigen Vectors pertaining to a matrix.

Code:

```
import numpy as np
from numpy.linalg import eig
a = np.array([[1, 1, 2],
              [1, 3, 1],
              [2,3, 1]])
w,v=eig(a)
print('E-value:', w)
print('E-vector', v)
```

Output:



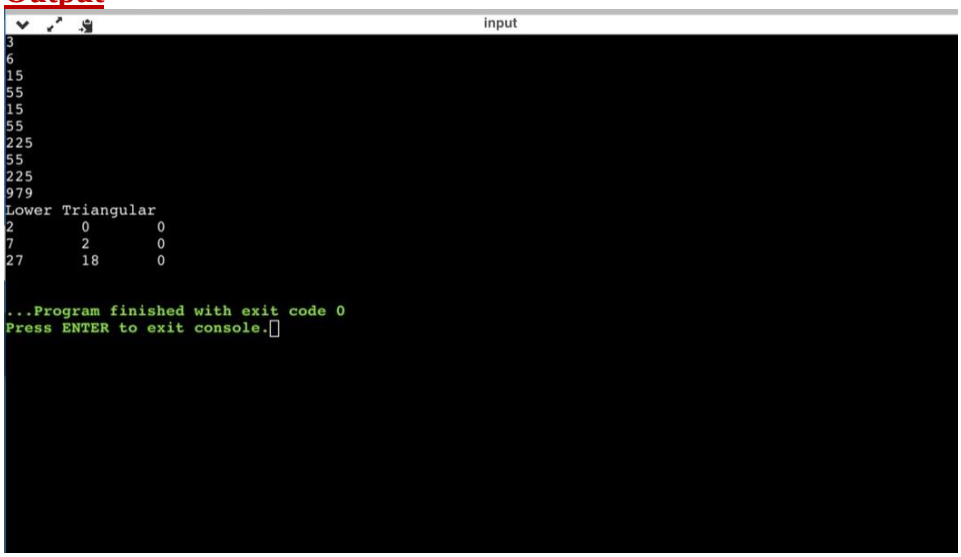
4. Write a program to reduce a matrix pertaining to a set of simultaneous equations into a lower triangle matrix and to solve it.

Code:

```
import java.util.*;
public class Main
{
public static void main(String[] args) throws Exception {
        Scanner sc = new Scanner(System.in);
        int n = sc.nextInt();
        int matrix[][] = new int[n][n];
        for (int i = 0; i < n; i++) {
            for (int j = 0; j < 3; j++) {
                matrix[i][j] = sc.nextInt();
            }
        }
        chMatrix(matrix, n);
        sc.close();

    }

    public static void chMatrix(int[][] matrix, int n) {
        int[][] lower = new int[n][n];
        for (int i = 0; i < n; i++) {
            for (int j = 0; j <= i; j++) {
                int sum = 0;
                if (j == i) {
                    for (int k = 0; k < j; k++)
                        sum += (int) Math.pow(lower[j][k], 2);

                    lower[j][j] = (int) Math.sqrt(matrix[j][j] - sum);
                } else {
                    for (int k = 0; k < j; k++)
                        sum += (lower[i][k] * lower[j][k]);

                    lower[i][j] = (matrix[i][j] - sum) / lower[j][j];
```

```
            }
        }

    }
    System.out.println("Lower Triangular \t ");
    for (int i = 0; i < n; i++) {
        for (int j = 0; j < n; j++) {
            System.out.print(lower[i][j] + "\t");
        }
        System.out.println();
        }
       }

}
```

Output



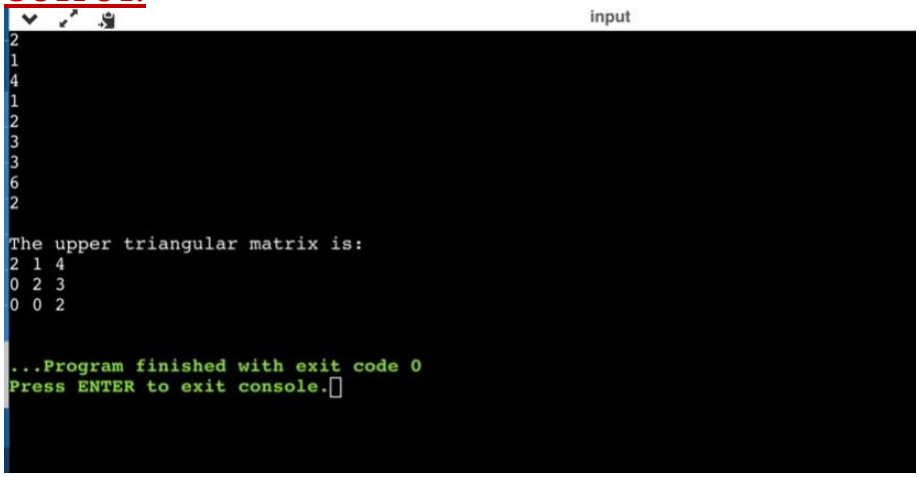
5. Write a program to reduce a matrix pertaining to a set of simultaneous equations into an upper triangle matrix and to solve it.

CODE:

```
import java.util.*;
public class Main
{public static void upper_triangular_matrix(int input_matrix[][]) {
  }
  public static void main(String[] args) {
    int input_matrix[][] = new int[3][3];
    Scanner sc=new Scanner(System.in);
   for(int i=0;i<3;i++)
   {
   for(int j=0;j<3;j++)
   {
```

```
    input_matrix[i][j]=sc.nextInt();
    }

    }
     int rows = input_matrix.length;
     int column = input_matrix[0].length;

     if (rows != column) {
       return;
     } else {
       for (int i = 0; i < rows; i++) {
         for (int j = 0; j < column; j++) {
           if (i > j) {
             input_matrix[i][j] = 0;
           }
         }
       }
       System.out.println("\nThe upper triangular matrix is: ");
       for (int i = 0; i < rows; i++) {
         for (int j = 0; j < column; j++) {
           System.out.print(input_matrix[i][j] + " ");
         }
         System.out.println();
       }
     }
   }
}
```

OUTPUT:



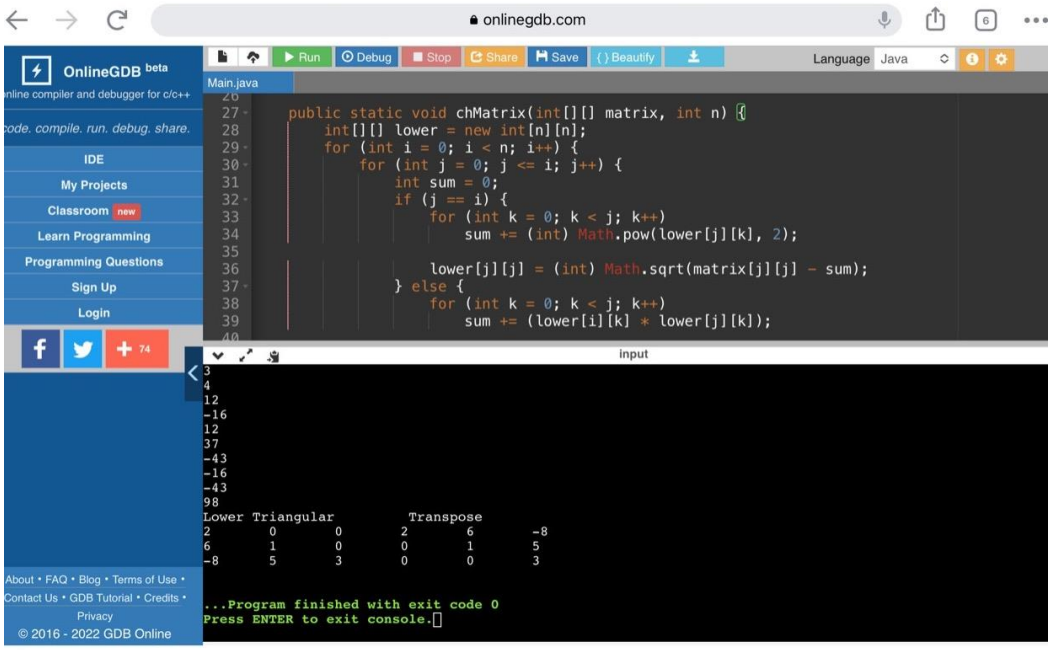
6. Write a program to find the Cholesky decomposition of a matrix.

Code:

```
import java.util.*;
public class Main
```

```
{
public static void main(String[] args) throws Exception {
        Scanner sc = new Scanner(System.in);
        int n = sc.nextInt();
        int matrix[][] = new int[n][n];
        for (int i = 0; i < n; i++) {
            for (int j = 0; j < 3; j++) {
                matrix[i][j] = sc.nextInt();
            }
        }
        chMatrix(matrix, n);
        sc.close();

    }

    public static void chMatrix(int[][] matrix, int n) {
        int[][] lower = new int[n][n];
        for (int i = 0; i < n; i++) {
            for (int j = 0; j <= i; j++) {
                int sum = 0;
                if (j == i) {
                    for (int k = 0; k < j; k++)
                        sum += (int) Math.pow(lower[j][k], 2);

                    lower[j][j] = (int) Math.sqrt(matrix[j][j] - sum);
                } else {
                    for (int k = 0; k < j; k++)
                        sum += (lower[i][k] * lower[j][k]);

                    lower[i][j] = (matrix[i][j] - sum) / lower[j][j];
                }
            }

        }
        System.out.println("Lower Triangular \t Transpose");
        for (int i = 0; i < n; i++) {
            for (int j = 0; j < n; j++) {
                System.out.print(lower[i][j] + "\t");
            }
            System.out.print("");
            for (int j = 0; j < n; j++) {
                System.out.print(lower[j][i] + "\t");
            }
            System.out.println();
        }
    }

}
```

Output:

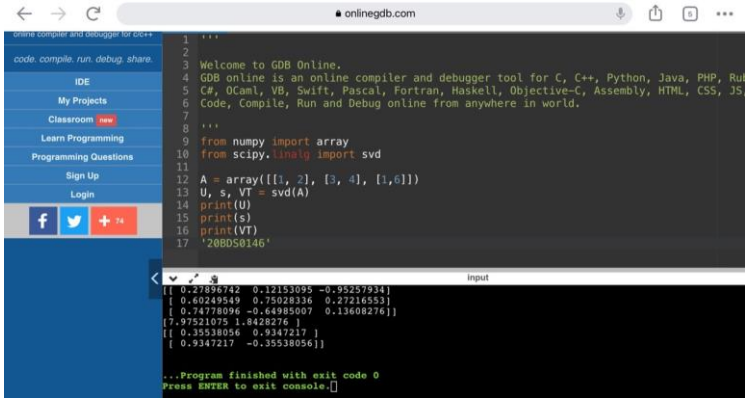


7. Write a program Perform Single Valued Decomposition of a matrix.

Code:

```
from numpy import array
from scipy.linalg import svd

A = array([[1, 2], [3, 4], [1,6]])
U, s, VT = svd(A)
print(U)
print(s)
print(VT)
```

Output:



8. Write a program to perform LU decomposition of

matrix.

Code:

```
import java.util.*;
public class Main
{
public static void main(String args[])
   {
      System.out.println("Enter dimension of the matrix:");
      Scanner sc = new Scanner(System.in);
      int n = sc.nextInt();
      double [][]mat = new double[n][n];
      for(int i=0; i<n; i++)
         for(int j=0; j<n; j++)
            mat[i][j] = sc.nextDouble();

      if(n==2)
      {
         double [][]l = new double[n][n];
         l[0][0] = l[1][1] = 1;
         l[0][1] = 0;

         double [][]u = new double[n][n];
         u[1][0] = 0;

         u[0][0] = mat[0][0];
         u[0][1] = mat[0][1];

         l[1][0] = mat[1][0]/mat[0][0];
         u[1][1] = mat[1][1] - (l[1][0]*u[0][1]);

         System.out.println("L=");
         for(int i=0; i<n; i++)
         {
            for(int j=0; j<n; j++)
               System.out.print(" "+l[i][j]);
            System.out.println();
         }
         System.out.println("The U=");
         for(int i=0; i<n; i++)
         {
            for(int j=0; j<n; j++)
               System.out.print(" "+u[i][j]);
            System.out.println();
         }

      }
      if(n==3)
      {
         double [][]l = new double[n][n];
         l[0][0] = l[1][1] = l[2][2] = 1;
         l[0][1] = l[0][2] = l[1][2] = 0;

         double [][]u = new double[n][n];
         u[1][0] = u[2][0] = u[2][1] = 0;

         u[0][0] = mat[0][0];
         u[0][1] = mat[0][1];
         u[0][2] = mat[0][2];

         l[1][0] = mat[1][0]/mat[0][0];
         u[1][1] = mat[1][1] - (l[1][0]*u[0][1]);
         u[1][2] = mat[1][2] - (l[1][0]*u[0][2]);

         l[2][0] = mat[2][0]/u[0][0];
         l[2][1] = (mat[2][1] - l[2][1]*u[0][1])/u[1][1];
         u[2][2] = mat[2][2] - (l[2][0]*u[0][2]) - (l[2][1]*u[1][2]);

         System.out.println("L=");
```

```
        for(int i=0; i<n; i++)
        {
            for(int j=0; j<n; j++)
                System.out.print(" "+l[i][j]);
            System.out.println();
        }
        System.out.println("U=");
        for(int i=0; i<n; i++)
        {
            for(int j=0; j<n; j++)
                System.out.print(" "+u[i][j]);
            System.out.println();
        }
    }
    sc.close();
  }
}
```

Output:

```
    {
        System.out.println("Enter dimension of
        Scanner sc = new Scanner(System.in);
        int n = sc.nextInt();
        double [][]mat = new double[n][n];
        for(int i=0; i<n; i++)
            for(int j=0; j<n; j++)
                mat[i][j] = sc.nextDouble();

        if(n==2)
        {
            double [][]l = new double[n][n];
            l[0][0] = l[1][1] = 1;
            l[0][1] = 0;

            double [][]u = new double[n][n];
            u[1][0] = 0;

            u[0][0] = mat[0][0];
            u[0][1] = mat[0][1];

            l[1][0] = mat[1][0]/mat[0][0];
            u[1][1] = mat[1][1] - (l[1][0]*u[0
            System.out.println("L=");
            for(int i=0; i<n; i++)
            {
                for(int j=0; j<n; j++)
                    System.out.print(" "+l[i][j
                System.out.println();
            }
            System.out.println("The U=");
            for(int i=0; i<n; i++)
            {
```

input

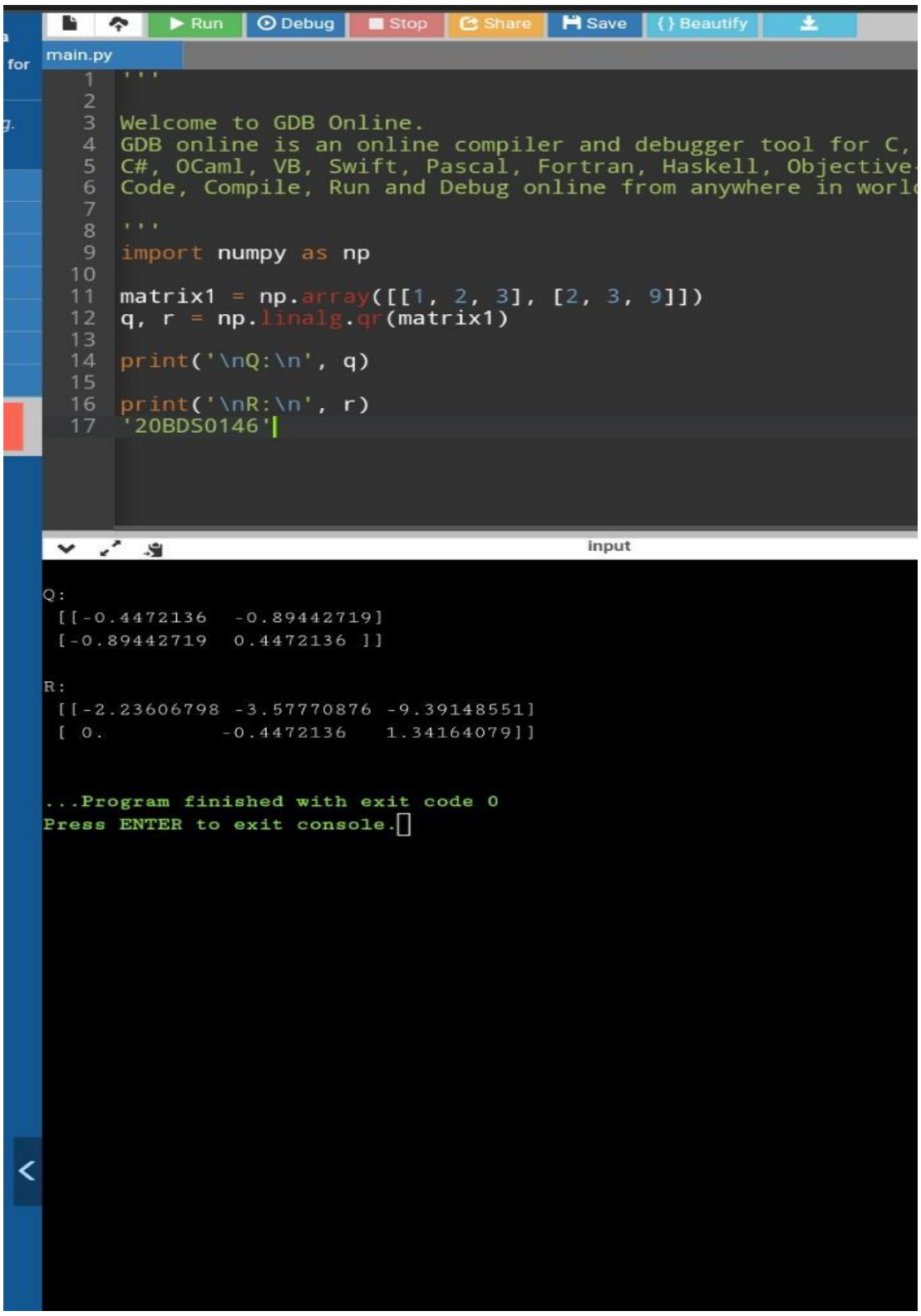
```
Enter dimension of the matrix:
3
1
2
1
3
2
4
1
5
2
L=
 1.0 0.0 0.0
 3.0 1.0 0.0
 1.0 -1.25 1.0
U=
 1.0 2.0 1.0
 0.0 -4.0 1.0
 0.0 0.0 2.25
```

9. Write a program to perform QR decomposition of matrix

Code:

```
import numpy as np

matrix1 = np.array([[1, 2, 3], [2, 3, 9]])
q, r = np.linalg.qr(matrix1)

print('\nQ:\n', q)
print('\nR:\n', r)
```

Output:



10. Write a program to find the Eigen decomposition of a matrix.

Code:

```
import numpy as np
def round(values, decs=0):
  return np.round(values*10*decs)/(10*decs)
A=[]
n=int(input("Enter size of Matrix : "))
print("Enter Values")
for i in range(n):
   row = []
   for j in range(n):
      row.append(int(input()))
   A.append(row)
Lambda, U = np.linalg.eig(A)
print("Eigen Vectors:")
print(U)
print("Eigen Values:")
print(Lambda)
inv_U = np.linalg.inv(U)
Λ = np.diag(Lambda)
vec = np.dot(U,np.dot(Λ, inv_U))

print("inverse of U:")
print(inv_U)
print("diagonal matrix with eigan value:")
print(Λ)
print("Eigen decomposition:")
print(vec)
```

Output: