

**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");}
}}

```

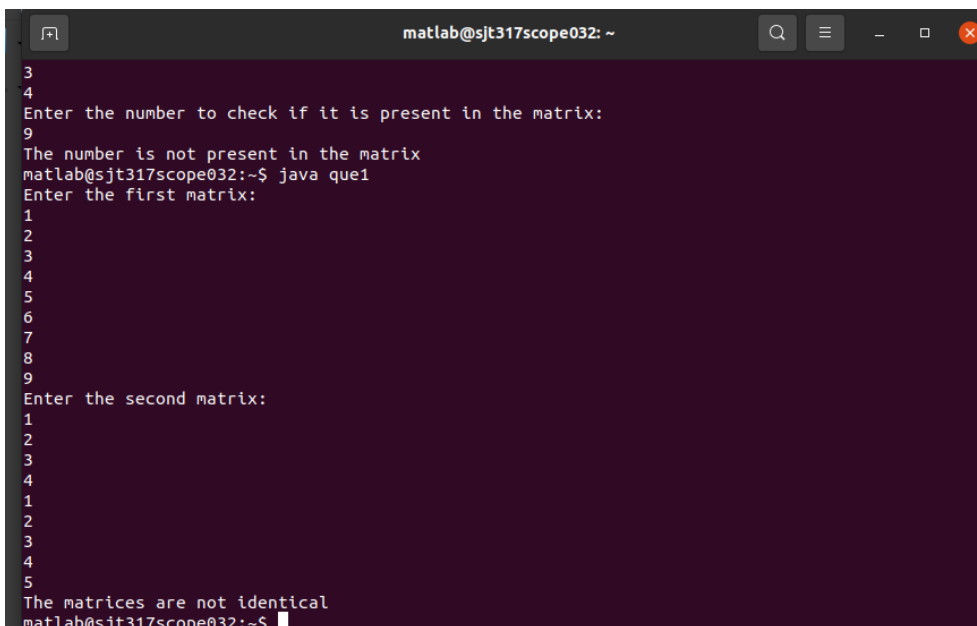
### Output:



```

matlab@sjt317scope032: ~
matlab@sjt317scope032:~$ javac que1.java
matlab@sjt317scope032:~$ java que1
Enter the first matrix:
1
2
3
4
5
6
7
8
9
Enter the second matrix:
1
2
3
4
5
6
7
8
9
The matrices are identical
matlab@sjt317scope032:~$

```



```

matlab@sjt317scope032: ~
3
4
Enter the number to check if it is present in the matrix:
9
The number is not present in the matrix
matlab@sjt317scope032:~$ java que1
Enter the first matrix:
1
2
3
4
5
6
7
8
9
Enter the second matrix:
1
2
3
4
1
2
3
4
5
The matrices are not identical
matlab@sjt317scope032:~$

```

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");
        }
    }
}

```

```

matlab@sjt317scope032: ~
Enter the first matrix:
1
2
3
4
5
6
1
2
3
Enter the number to check if it is present in the matrix:
3
The number is present in the matrix
matlab@sjt317scope032:~$ java que2
Enter the first matrix:
1
2
3
4
5
1
2
3
4
Enter the number to check if it is present in the matrix:
9
The number is not present in the matrix
matlab@sjt317scope032:~$

```

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:**

The screenshot shows a web browser window with the URL 'onlinegdb.com'. The page has a sidebar with navigation links: IDE, My Projects, Classroom, Learn Programming, Programming Questions, Sign Up, and Login. The main area contains a Python code editor with the following code:

```

3 Online Python Compiler.
4 Code, Compile, Run and Debug python program online.
5 Write your code in this editor and press "Run" button to execute it.
6
7
8 import numpy as np
9 from numpy.linalg import eig
10 a = np.array([[1, 1, 2],
11              [1, 3, 1],
12              [2, 3, 1]])
13 w,v=eig(a)
14 print('E-value:', w)
15 print('E-vector', v)
16
17 '20BDS0146'

```

Below the code editor, the output is displayed:

```

E-value: [ 5. -1.  1.]
E-vector [[ 4.76731295e-01  7.07106781e-01  8.01783726e-01]
 [ 5.72077554e-01  2.75020947e-17 -5.34522484e-01]
 [ 6.67423812e-01 -7.07106781e-01  2.67261242e-01]]

```

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];
                }
            }
        }
    }
}

```

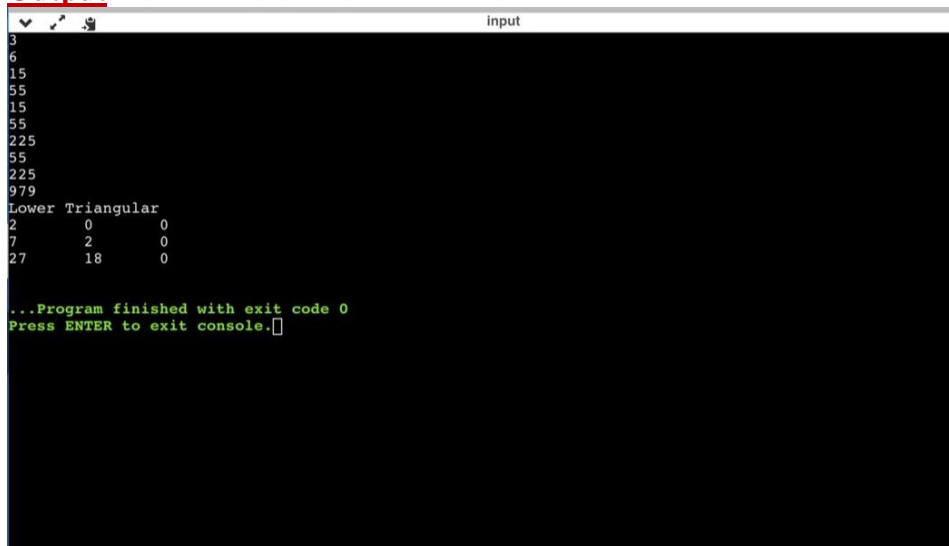
```

    }
}

}
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



```

3
6
15
55
15
55
225
55
225
979
Lower Triangular
2      0      0
7      2      0
27     18     0

...Program finished with exit code 0
Press ENTER to exit console.

```

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++)
            {

```

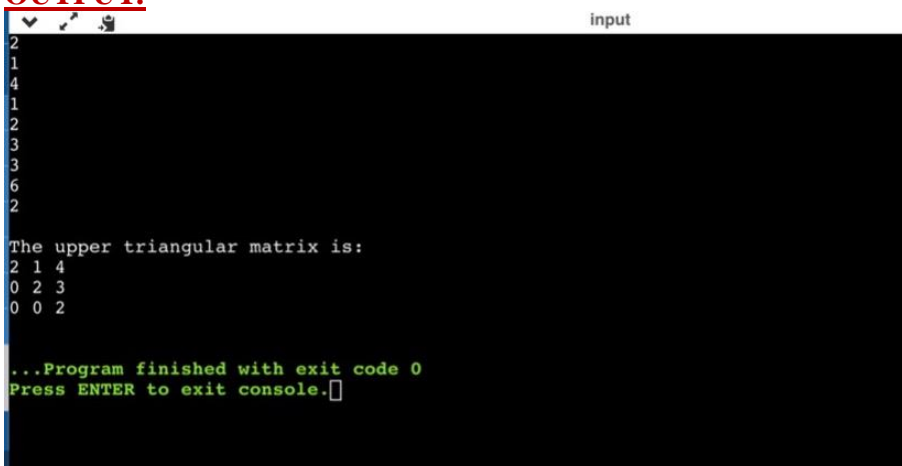
```

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:**


```

input
2
1
4
1
2
2
3
3
6
2
3
2
The upper triangular matrix is:
2 1 4 1
0 2 3 3
0 0 2 2
0 0 0 1
...Program finished with exit code 0
Press ENTER to exit console.

```

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:**



```

27- public static void chMatrix(int[][] matrix, int n) {
28-     int[][] lower = new int[n][n];
29-     for (int i = 0; i < n; i++) {
30-         for (int j = 0; j <= i; j++) {
31-             int sum = 0;
32-             if (j == i) {
33-                 for (int k = 0; k < j; k++)
34-                     sum += (int) Math.pow(lower[j][k], 2);
35-                 lower[j][j] = (int) Math.sqrt(matrix[j][j] - sum);
36-             } else {
37-                 for (int k = 0; k < j; k++)
38-                     sum += (lower[i][k] * lower[j][k]);
39-             }
40-         }
41-     }
42- }

```

Input:

```

3
4
12
-16
12
37
-43
-16
-43
98
Lower Triangular      Transpose
2      0      0      2      6      -8
6      1      0      0      1      5
-8     5      3      0      0      3

```

...Program finished with exit code 0  
Press ENTER to exit console.

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:

```

1 ...
2
3 Welcome to GDB Online.
4 GDB online is an online compiler and debugger tool for C, C++, Python, Java, PHP, Ruby,
5 C#, OCaml, VB, Swift, Pascal, Fortran, Haskell, Objective-C, Assembly, HTML, CSS, JS,
6 Code, Compile, Run and Debug online from anywhere in world.
7 ...
8
9 from numpy import array
10 from scipy.linalg import svd
11
12 A = array([[1, 2], [3, 4], [1,6]])
13 U, s, VT = svd(A)
14 print(U)
15 print(s)
16 print(VT)
17 '20BDS0146'

```

Output:

```

[[ 0.27896742  0.12153095 -0.95257934]
 [ 0.60249549  0.75028336  0.27216553]
 [ 0.74778096 -0.64895007  0.13608276]]
[ 7.97521075  1.8428276 ]
[[ 0.35538056  0.9347217 ]
 [ 0.9347217  -0.35538056]]

```

...Program finished with exit code 0  
Press ENTER to exit console.

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][0]*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:

```

12 {
13     System.out.println("Enter dimension of
14     Scanner sc = new Scanner(System.in);
15     int n = sc.nextInt();
16     double [][]mat = new double[n][n];
17     for(int i=0; i<n; i++)
18         for(int j=0; j<n; j++)
19             mat[i][j] = sc.nextDouble();
20
21     if(n==2)
22     {
23         double [][]l = new double[n][n];
24         l[0][0] = l[1][1] = 1;
25         l[0][1] = 0;
26
27         double [][]u = new double[n][n];
28         u[1][0] = 0;
29
30         u[0][0] = mat[0][0];
31         u[0][1] = mat[0][1];
32
33         l[1][0] = mat[1][0]/mat[0][0];
34         u[1][1] = mat[1][1] - (l[1][0]*u[0][1]);
35
36         System.out.println("L=");
37         for(int i=0; i<n; i++)
38         {
39             for(int j=0; j<n; j++)
40                 System.out.print(" "+l[i][j]);
41             System.out.println();
42         }
43         System.out.println("The U=");
44         for(int i=0; i<n; i++)
45         {
46             for(int j=0; j<n; j++)

```

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

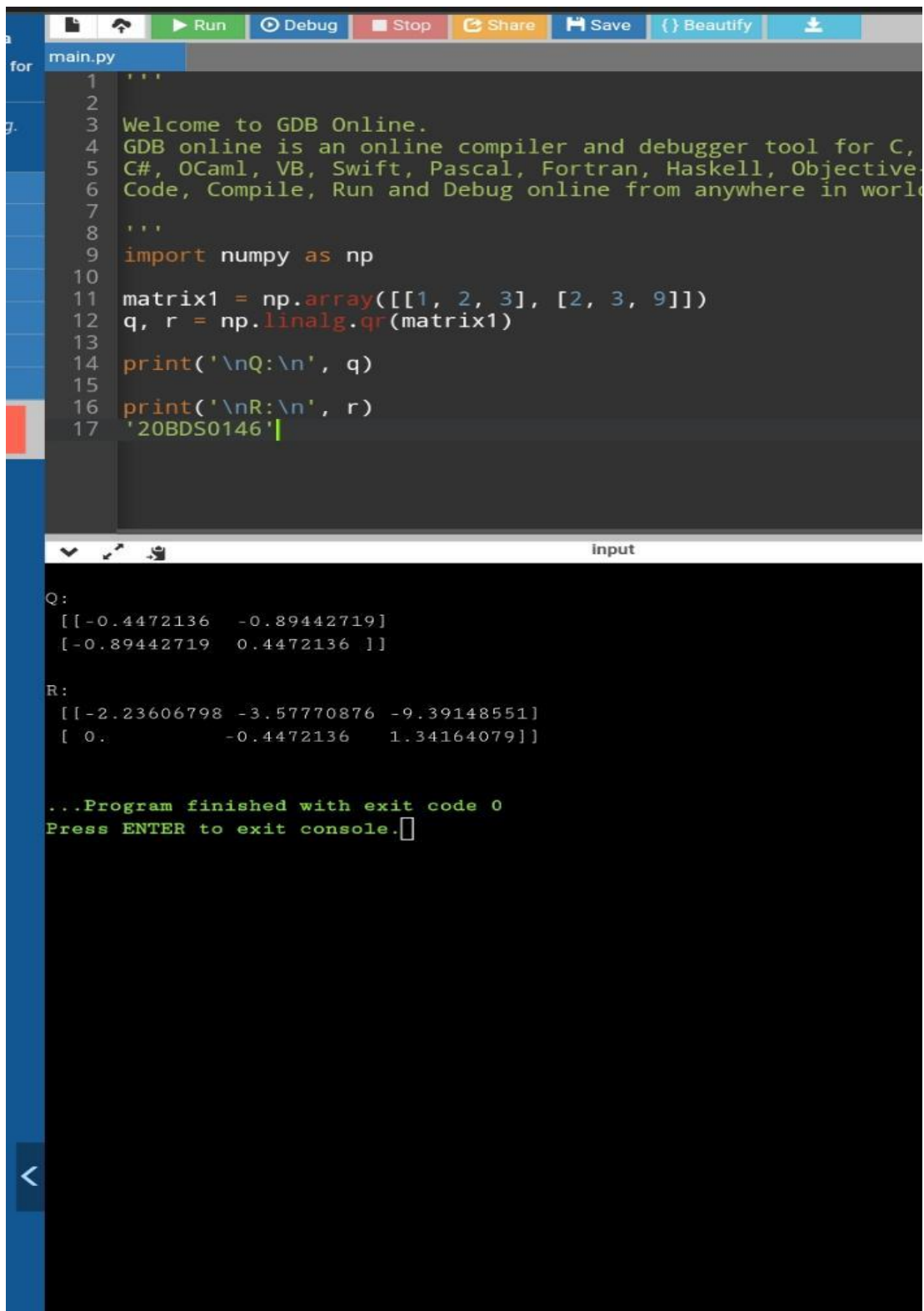
**Code:**

```
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:**



The screenshot shows a web-based IDE interface for GDB Online. At the top, there is a toolbar with buttons for Run, Debug, Stop, Share, Save, Beautify, and a download icon. Below the toolbar, the file name 'main.py' is displayed. The code editor contains a Python script that calculates the QR decomposition of a matrix. The script includes a welcome message, imports numpy, defines a 2x3 matrix, performs the QR decomposition using np.linalg.qr, and prints the resulting Q and R matrices. The output window at the bottom shows the execution results, displaying the Q and R matrices and a message indicating the program finished successfully.

```
1 '''
2
3 Welcome to GDB Online.
4 GDB online is an online compiler and debugger tool for C,
5 C#, OCaml, VB, Swift, Pascal, Fortran, Haskell, Objective-
6 Code, Compile, Run and Debug online from anywhere in world
7
8 '''
9 import numpy as np
10
11 matrix1 = np.array([[1, 2, 3], [2, 3, 9]])
12 q, r = np.linalg.qr(matrix1)
13
14 print('\nQ:\n', q)
15
16 print('\nR:\n', r)
17 '20BDS0146'
```

input

Q:

```
[[ -0.4472136  -0.89442719]
 [ -0.89442719   0.4472136 ]]
```

R:

```
[[ -2.23606798 -3.57770876 -9.39148551]
 [  0.          -0.4472136   1.34164079]]
```

...Program finished with exit code 0  
Press ENTER to exit console.

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

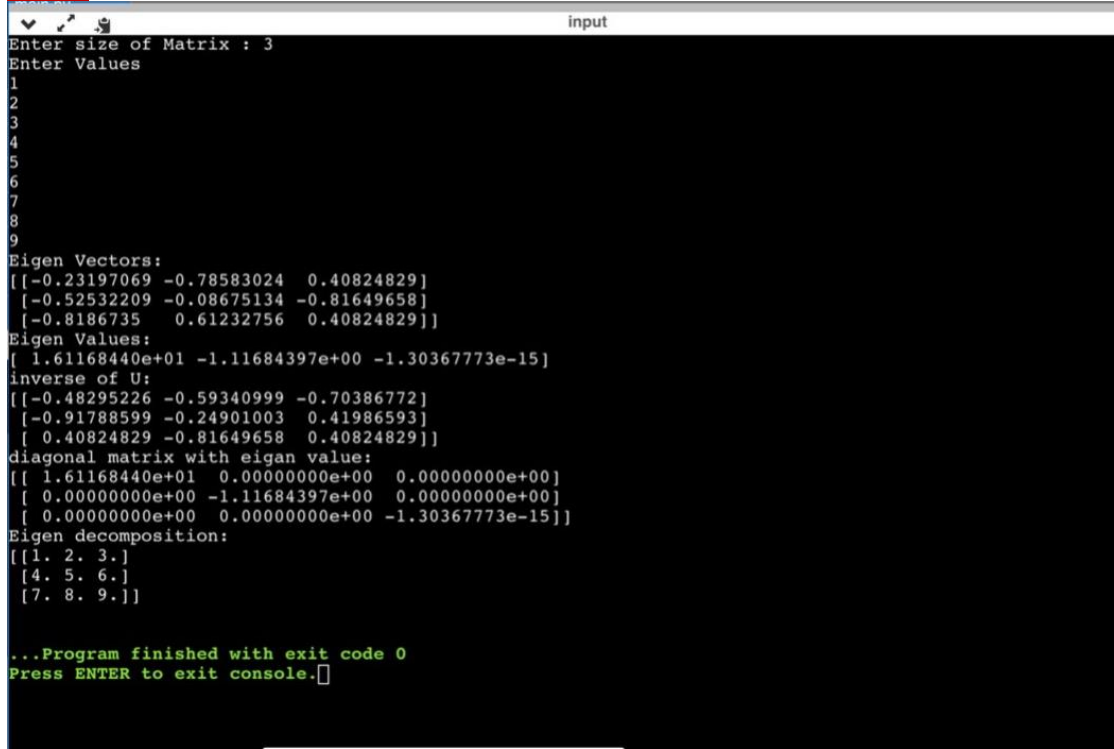
Code:

```

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 eigen value:")
print(Λ)
print("Eigen decomposition:")
print(vec)

```

Output:


```

input
Enter size of Matrix : 3
Enter Values
1
2
3
4
5
6
7
8
9
Eigen Vectors:
[[-0.23197069 -0.78583024  0.40824829]
 [-0.52532209 -0.08675134 -0.81649658]
 [-0.8186735  0.61232756  0.40824829]]
Eigen Values:
[ 1.61168440e+01 -1.11684397e+00 -1.30367773e-15]
inverse of U:
[[-0.48295226 -0.59340999 -0.70386772]
 [-0.91788599 -0.24901003  0.41986593]
 [ 0.40824829 -0.81649658  0.40824829]]
diagonal matrix with eigen value:
[[ 1.61168440e+01  0.00000000e+00  0.00000000e+00]
 [ 0.00000000e+00 -1.11684397e+00  0.00000000e+00]
 [ 0.00000000e+00  0.00000000e+00 -1.30367773e-15]]
Eigen decomposition:
[[1. 2. 3.]
 [4. 5. 6.]
 [7. 8. 9.]]

...Program finished with exit code 0
Press ENTER to exit console.

```