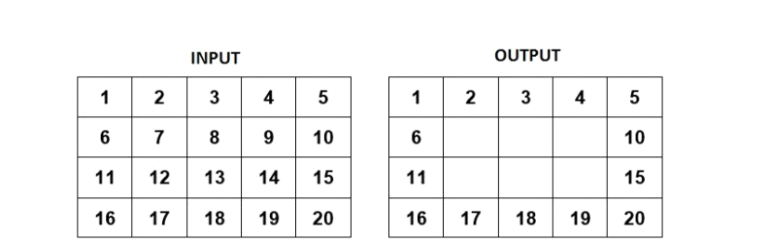
**ARRAY**

**PROGRAMS**

PROGRAM 1

Write a Program in Java to input a 2-D array of size ‘m\*n’ and print its boundary (border) elements. For example:



165

Algorithm

Step 1: Start.

Step 2: Scanner class object is declared.

Step 3: The number of rows and the number of columns are taken as input in variable m and n respectively.

Step 4: An array arr with m rows and n columns is declared.

Step 5: The elements of the array are taken from the user.

Step 6: The array is then printed.

Step 7: An outer for loop starts which terminates when its counter variable i is less than m.

Step 8: An inner loop starts which terminates if its counter variable j becomes less than n.

Step 9: If i or j equals to zero or n-1 ,element of the array of ith row and jth column is printed.

Step 10: Stop.

166

Source Code



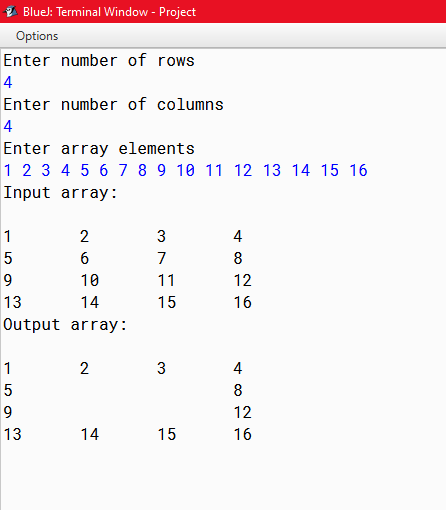
167

Variable Description

|  |  |  |
| --- | --- | --- |
| Variable name | Data type | Description |
| m | Int | To enter the number of rows by the user |
| n | Int | To enter the number of columns by the user |
| arr[][] | Int | Double dimensional array to enter the number of rows and columns |
| i | Int | Loop variable |
| j | int | Loop variable |

168

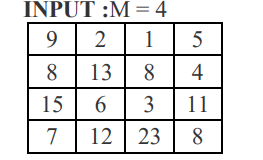
OUTPUT

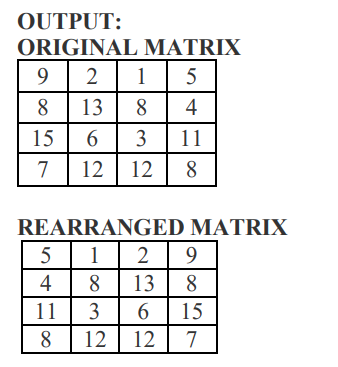


169

PROGRAM 2

Write a program to declare a square matrix A[][] of order (M x M) where ‘M’ must be greater than 3 and less than 10. Allow the user to input positive integers into this matrix. Display the matrix in such a way that the resultant matrix appears to be the mirror image of the original matrix.





170

Algorithm

Step 1: Start.

Step 2: A class named mirror\_array is declared.

Step 3: public static void main.

Step 4: Scanner class object is declared.

Step 5: The number of rows is taken from user provided it is greater than 2 and less 10 for the square matrix.

Step 6: If m is greater than 10 or less 2 invalid output.

Step 7: An array a of size m\*m is declared.

Step 8: All the elements of the matrix are taken from the user.

Step 9: An array b of size m\*m is declared.

Step 10: An outer for loop starts which terminates if its counter variable i is equal to m.

Step 11: An inner loop starts which terminates if its counter variable J is equal to m.

Step 12: A variable k is initialized with 3 outside the inner for loop.

Step 13: For each execution of the loop element of array b of ith row and kth column is initialized with the value of a[i][j].

Step 14: The matrix b which is formed is printed.

Step 15: Stop.

171

Source Code



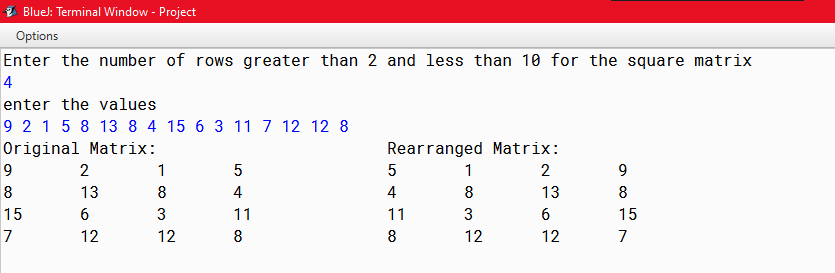
172

Variable Description

|  |  |  |
| --- | --- | --- |
| Variable name | Data type | Description |
| m | Int | To enter the number of rows by the user |
| b[][] | Int | Double dimensional array to enter the number of rows and columns |
| a[][] | Int | Double dimensional array to enter the number of rows and columns |
| i | Int | Loop variable |
| j | int | Loop variable |
| k | Int | Flag variable |

173

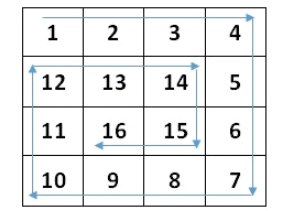
OUTPUT



174

PROGRAM 3

Write a Program in Java to fill a square matrix of size ‘n\*n” in a circular fashion (clockwise) with natural numbers from 1 to n\*n, taking ‘n’ as input. For example: if n = 4, then n\*n = 16, hence the array will be filled as given below.



175

Algorithm

Step 1: start

Step 2: scanner class object declared

Step 3: enter the number of rows and columns in m variable

Step 4: store m to n variable

Step 5: create an array a[][] of (MAX\*MAX) size

Step 6: call the function spiralFill() by passing the values of m, n, a variables.

Step 7: display the result by printing a[i][j],where i and j are loop variables

Step 8: Stop.

Algorithm for spiralFill()

Step 1: start

Step 2: the number passed from the main function is stored in m ,n, a[][] variables respectively

Step 3: initialize val =1, k=0, l=0;

Step 4: Run a while loop which will be executed only when k is less than m and l is less than n

Step 4.1: run a for loop i from l to n. store val++ value in a[k][i]. after execution of the for loop increment k variable by 1

Step 4.2: run another for loop i from k to m. store val++ value in a[i][n-1]. after execution of the for loop decrement n variable by 1

Step 4.3: check if k is less than m or not. If it is then run the loop i from n-1 to l(backward).store thevalue of val++ in a[m-1][i]. after execution of the for loop decrement m variable by 1

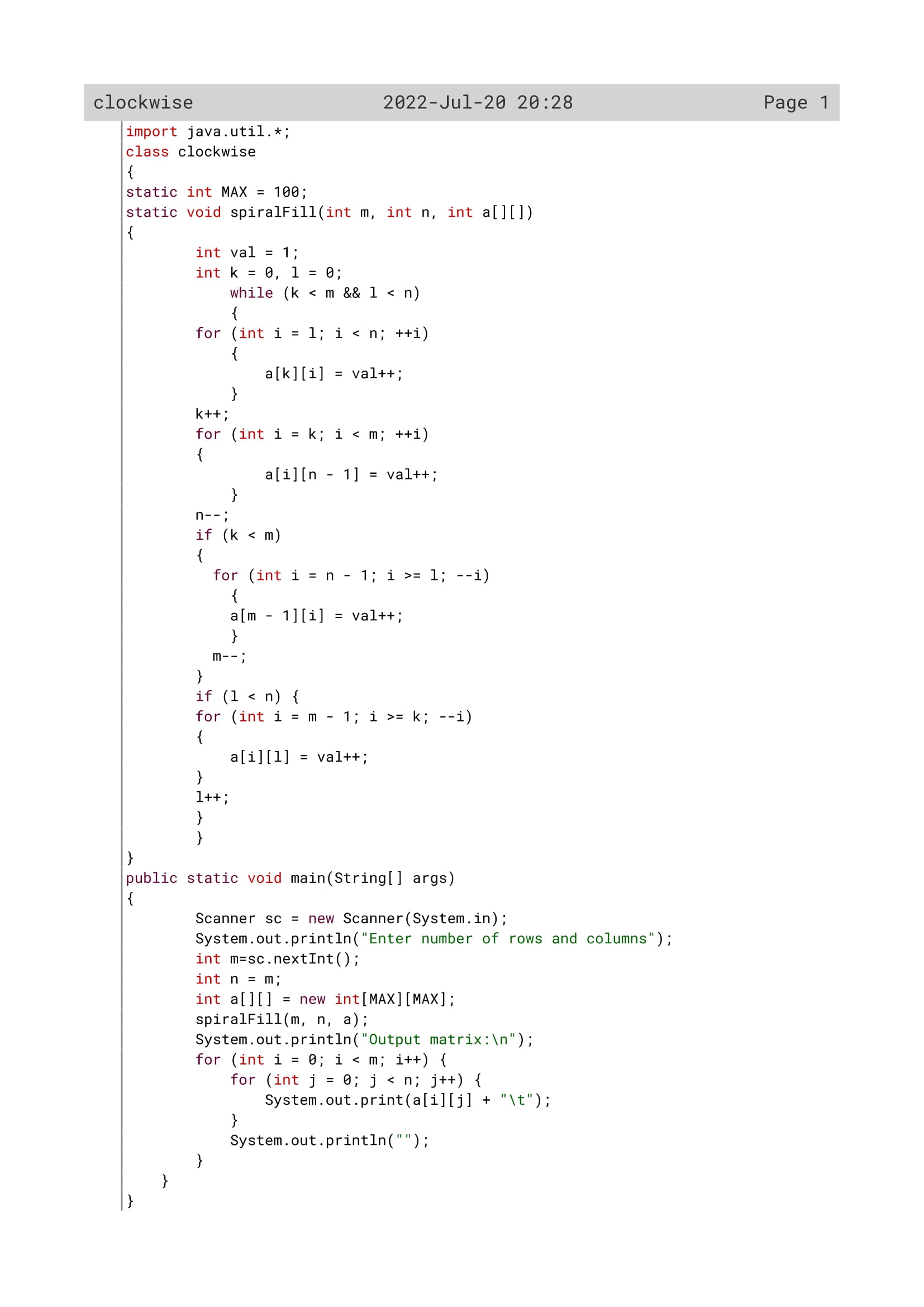
Step 4.4: check if l is less than n or not. If it is then run the loop i from m-1 to k(backward).store thevalue of val++ in a[i][l]. after execution of the for loop increment l variable by 1

step 5: In this way the number will be arranged in circular fashion clockwise.

Step 6: Stop

176

Source Code



177

Variable Description

|  |  |  |
| --- | --- | --- |
| Variable name | Data type | Description |
| MAX | Int | Static variable |
| i | Int | Loop variable |
| j | Int | Loop variable |
| val | Int |  |
| m | Int | To enter the number of rows by the user |
| n | Int | To enter the number of columns by the user |
| k | Int | Flag variable |
| a[][] | Int | To enter the elements in a double dimensional array |
| l | int | Flag variable |

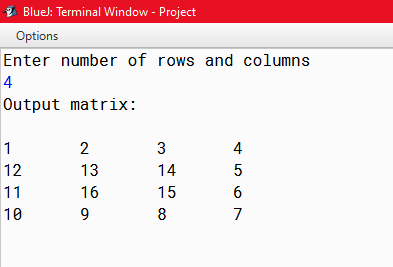
178

Function Description

|  |  |  |
| --- | --- | --- |
| Function name | Data type | description |
| main() | void | To accept values and display the array and call spiralfill(). |
| spiralfill() | void | To manipulate the array and fill it with numbers in spiral fashion. |

179

OUTPUT



180

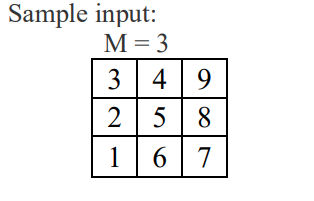
PROGRAM 4

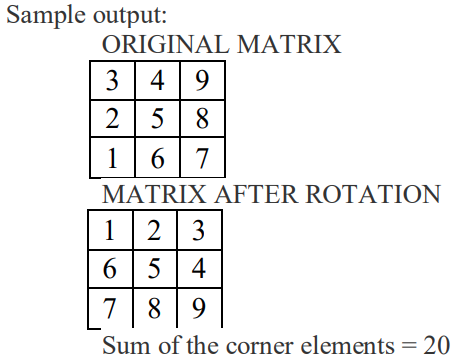
Write a program to declare a square matrix A[ ][ ] of order MxM where ‘M’ is the number of rows and the number of columns, such that M must be greater than 2 and less than 10. Accept the value of M as user input. Display an appropriate message for an invalid input. Allow the user to input integers into this matrix. Perform the following tasks:

(a) Display the original matrix.

(b) Rotate the matrix 90° clockwise as shown below:

(c) Find the sum of the elements of the four corners of the matrix.





181

Algorithm

Step 1: Start.

Step 2: Scanner class object is declared.

Step 3: The number of between 3 and 10.

Step 4: check if the number of rows entered is less than 3 or greater than 10 or not. If the condition satisfies then print “size out of range”.

Step 5: if the above condition does not satisfies then a 2-D array *a* of order [m][m] and a single dimensional array *b* of size [m\*m] is declared.

Step 6: Enter the elements of the array using i and j loop variable and store it in *a* variable.

Step 7: a ctr variable is declared and initialize to 0;

Step 8: print the array *a* and also initialize b[crt++]=a[i][j];

Step 9: declare c variable and initialize it to m-1, and ctr=0;

Step 10: Now in order to rotate the matrix by 90 degree run a do-while loop unless c is equal to 0 and a for loop i from 0 to less than m. store the elements of a[i][c]=b[ctr++] and decrease the variable c by 1;

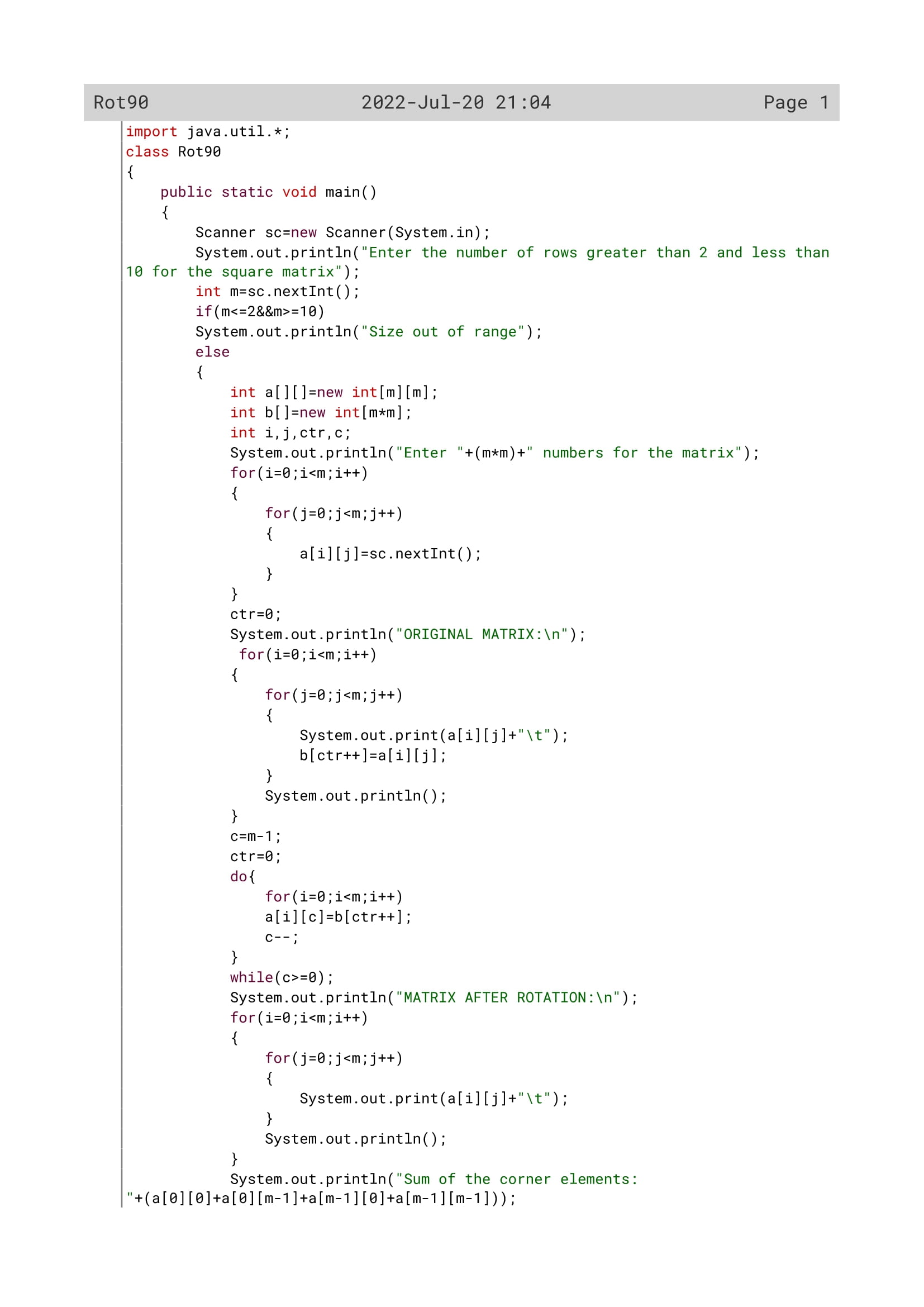
Step 11: print the new matrix which is rotated 90 degree clockwise.

Step 12: Now print the sum of the array by adding the corner elements of array *a* (a[0][0]+a[0][m-1]+a[m-1][0]+a[m-1][m-1];

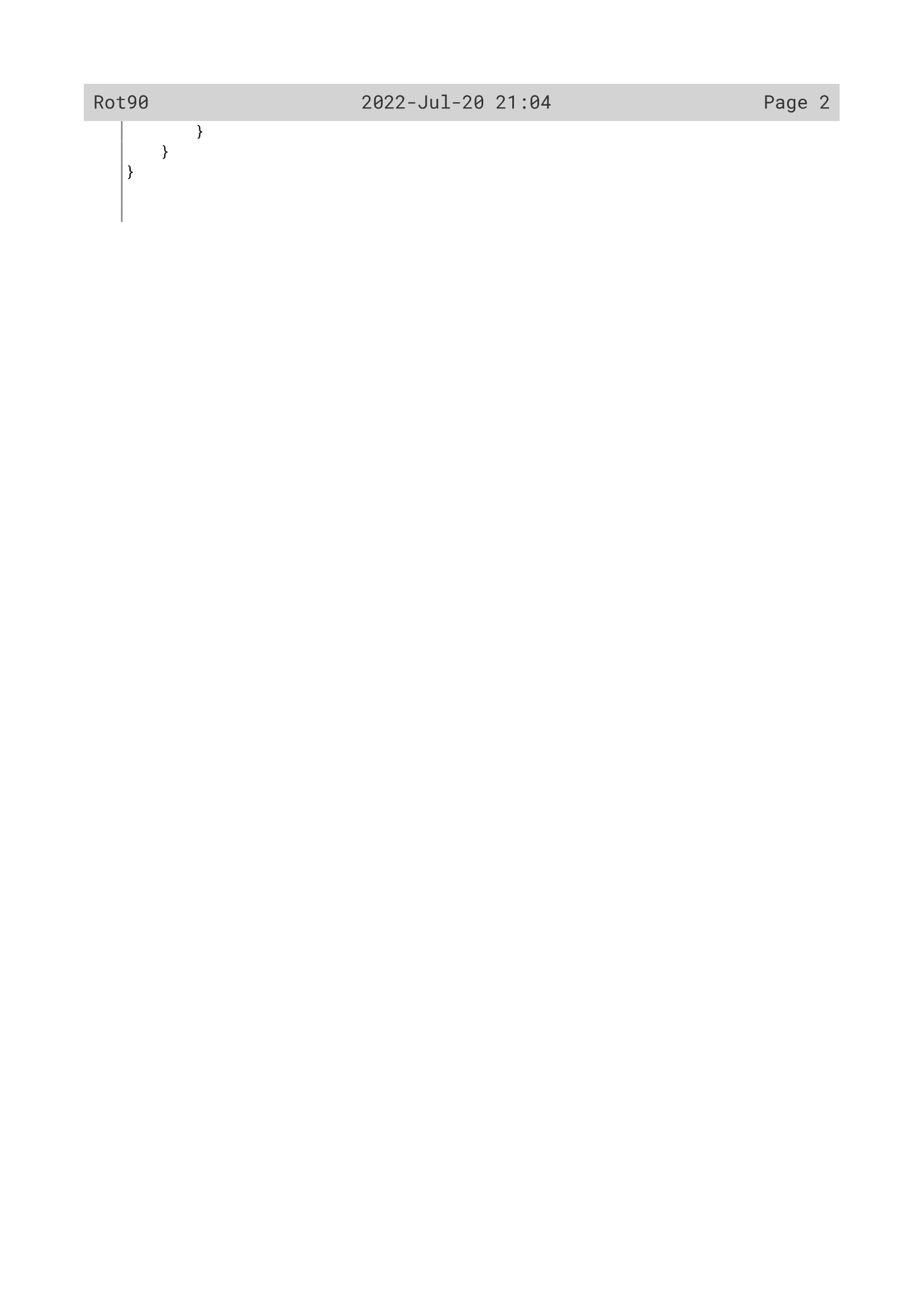
Step 13: Stop

182

Source Code



183



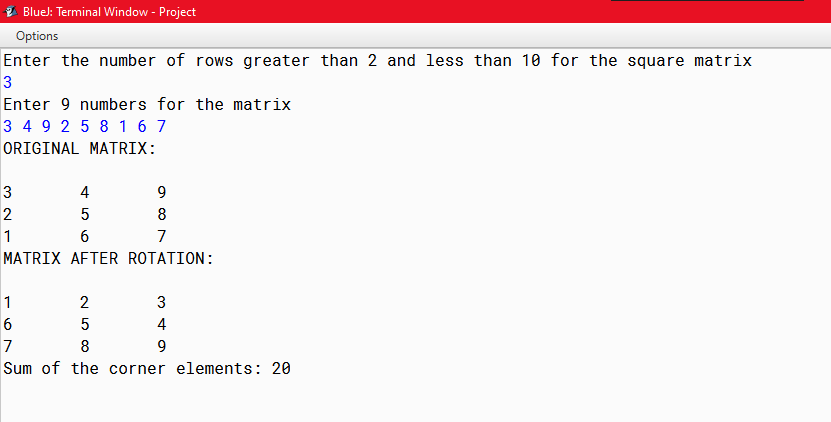
184

Variable Description

|  |  |  |
| --- | --- | --- |
| Variable name | Data type | Description |
| m | Int | To enter the number of rows by the user |
| b[] | Int | Single dimensional array to calculate multiplication of two rows |
| arr[][] | Int | Double dimensional array to enter the number of rows |
| i | Int | Loop variable |
| j | int | Loop variable |
| c | Int | Flag variable |
| ctr | Int | Flag variable |

185

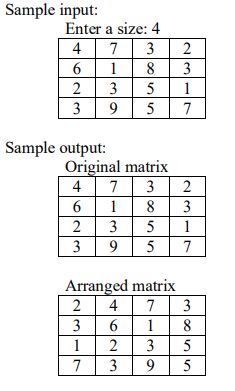
OUTPUT



186

PROGRAM 5

Write a program in java to enter natural numbers in a 2D array. Shift the elements of 4th column into the 1st column. Elements of 1st column into the 2nd column & so on.



187

Algorithm

Step 1: Start.

Step 2: Scanner class object is declared.

Step 3: The number of rows and the number of columns are taken as input in variable m and n respectively.

Step 4: An array arr with m rows and n columns is declared.

Step 5: An array newarr with m rows and n columns is declared.

Step 6: the array elements are taken from user using two loops i and j

Step 7: the array elements are printed using two loops i and j

Step 8: the j loop is executed from 0 to number of columns.

Step 9: In col variable initialize j+1;

Step 10: check if col is equal to n or not.if equal then store 0 to col variable.

Step 11: now run the loop i to shift the elements of the 4rt column in 1st column, elements of 1st column to 2nd column and so on.

Step 12: print the array newarr.

Step 13: Stop.

188

Source Code



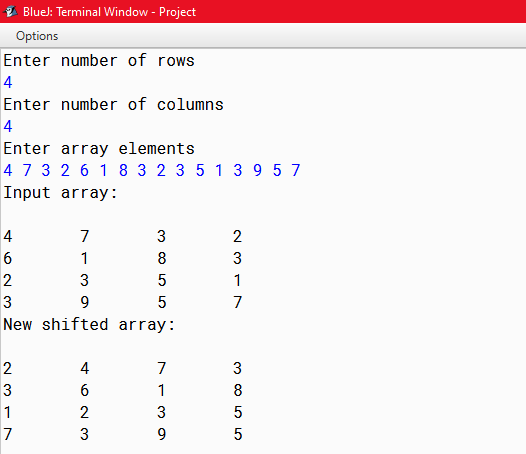
189

Variable Description

|  |  |  |
| --- | --- | --- |
| Variable name | Data type | Description |
| m | Int | To enter the number of rows by the user |
| n | Int | To enter the number of columns by the user |
| arr[][] | Int | Double dimensional array to enter the number of rows and columns |
| i | Int | Loop variable |
| newarr[] | Int | To display the new shifted array to the user |
| j | int | Loop variable |
| Col | Int | To shift the column |

190

OUTPUT



191

PROGRAM 6

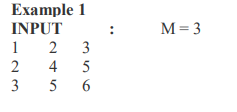
Write a program to declare a square matrix A[ ] [ ] of order (M x M) where ‘M’ is the number of rows and the number of columns such that M must be greater than 2 and less than 10. Accept the value of M as user input. Display an appropriate message for an invalid input. Allow the user to input integers into this matrix. Perform the following tasks:

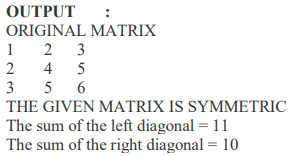
(a) Display the original matrix.

(b) Check if the given matrix is Symmetric or not. A square matrix is said to be Symmetric, if the element of the ith row and jth column is equal to the element of the jth row and ith column.

(c) Find the sum of the elements of left diagonal and the sum of the elements of right diagonal of the matrix and display them.

Test your program with the sample data and some random data:





192

Algorithm

Step 1: Start.

Step 2: A class named symmetric is declared.

Step 3: public static void main.

Step 4: Scanner class object is declared.

Step 5: The number of rows is taken from user provided it is greater than 2 and less 10 for the square matrix.

Step 6: If m is greater than 10 or less than 2 it gives invalid output.

Step 7: An array arr of size m\*m is declared.

Step 8: All the elements of the matrix are taken from the user.

Step 9: A flag variable is initialized with 0.

Step 10: An outer for loop starts which terminates if its counter variable i is equal to m.

Step 11: An inner loop starts which terminates if its counter variable J is equal to m.

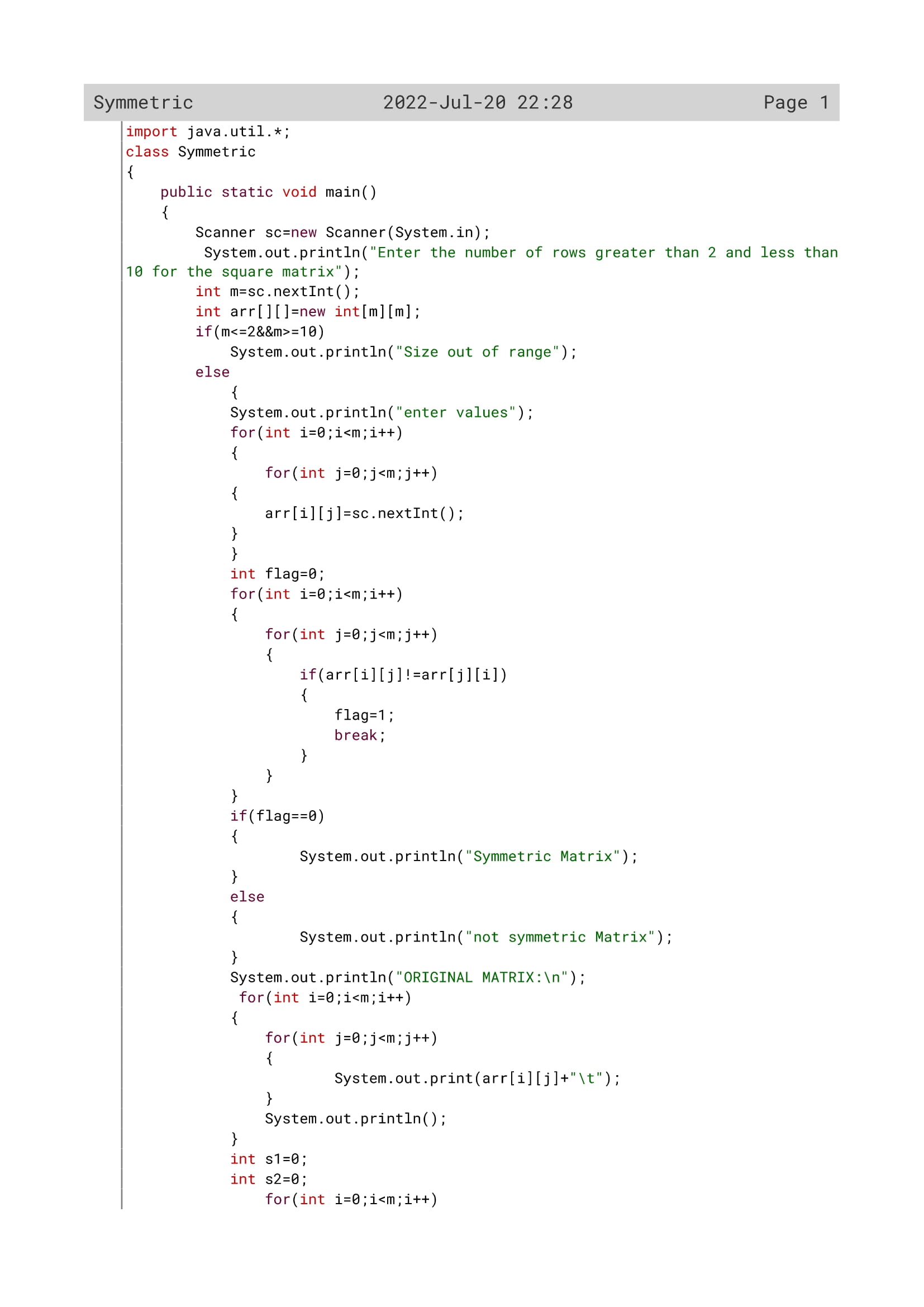
Step 12: If element of the ith row and jth column of matrix arr is not equal to element of the jth row and ith column of matrix arr ,flag becomes 1 and break statement is executed.

Step 13: If flag is 0 the matrix is printed symmetric else it is not.

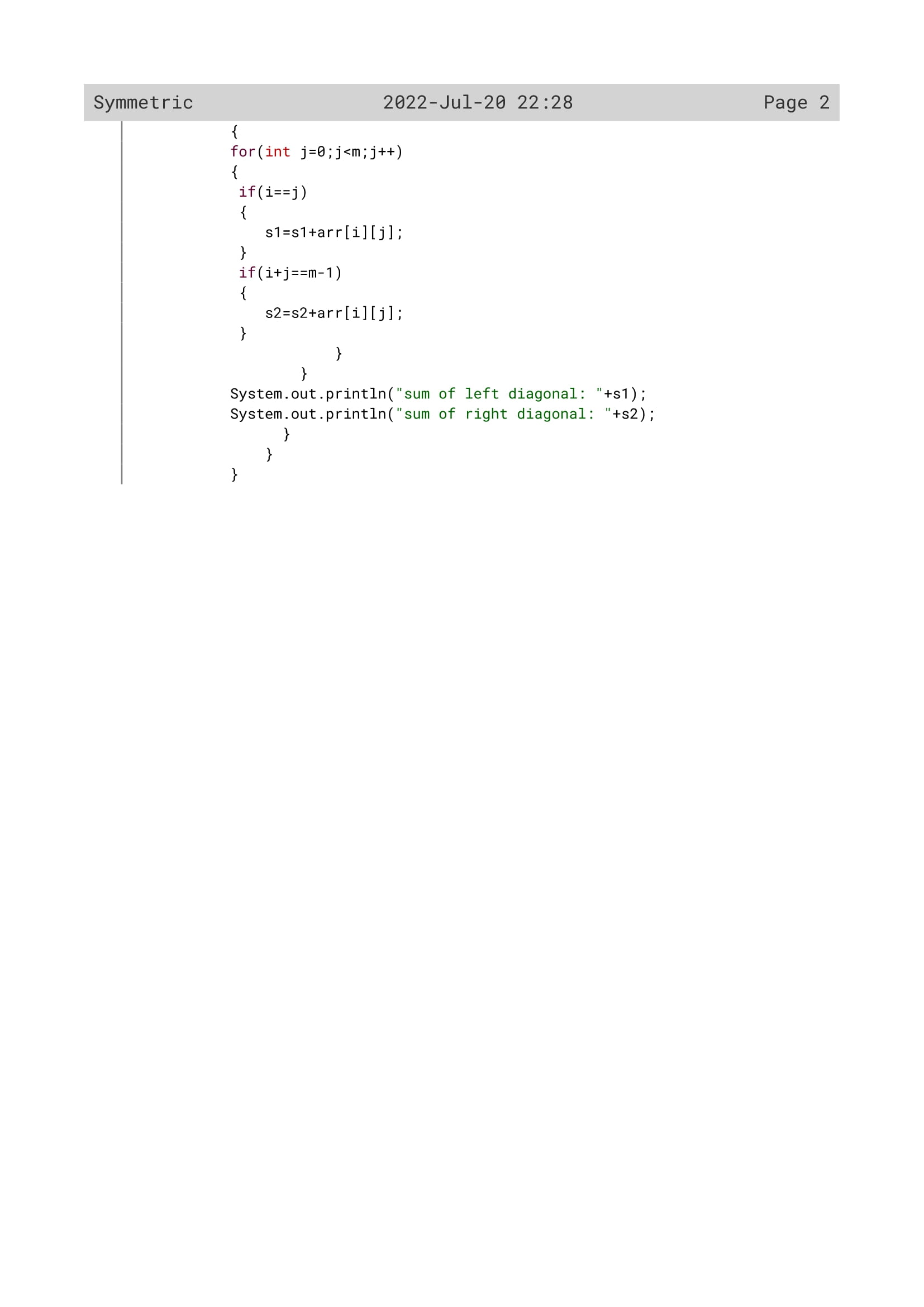
Step 14: Stop.

193

Source Code



194



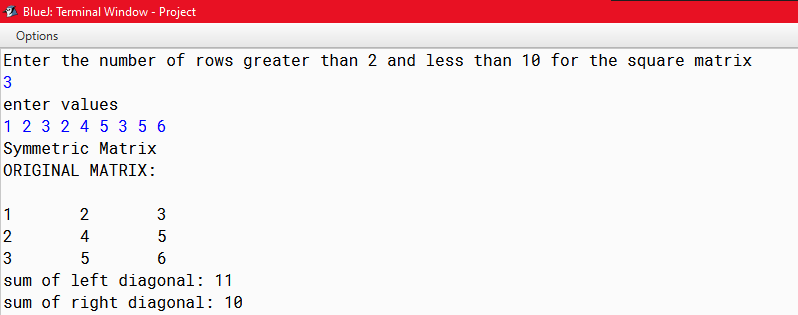
195

Variable Description

|  |  |  |
| --- | --- | --- |
| Variable name | Data type | Description |
| m | Int | To enter the number of rows by the user |
| Flag | Int | Flag variable |
| arr[][] | Int | Double dimensional array to enter the number of rows |
| i | Int | Loop variable |
| s1 | int | To calculate and display the sum first diagonal to the user |
| j | Int | Loop variable |
| s2 | Int | To calculate and display the sum 2 diagonal to the user |

196

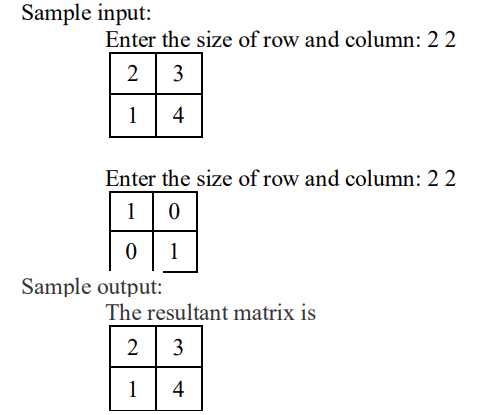
OUTPUT



197

PROGRAM 7

Write a program in java to store elements in two different matrix & find the product of two matrices & store the result in a third matrix.(size of row and column can be different. But the size of column of 1st matrix must be equal to the size of row of 2nd matrix)



198

Algorithm

Step 1: Start.

Step 2: Scanner class object is declared.

Step 3: the number of rows of the first matrix is accepted from the user and stored in R1 variable

Step 4: the number of columns of the first matrix is accepted from the user and stored in C1 variable

Step 5: the number of rows of the second matrix is accepted from the user and stored in R2 variable

Step 6: the number of columns of the second matrix is accepted from the user and stored in C2 variable

Step 7: An array A of size [R1][C1] is declared.

Step 8: An array B of size [R2][C2] is declared.

Step 9: An array D of size[5][5] is declared.

Step 10: the elements of the first matrix and second matrix is accepted from the user and are printed.

Step 11: check if C1==R2 or not.

Step 12: if the condition satisfies run a loop i from 0 to less than R2 and an inner loop j from 0 to C1.

Step 13: Initialize D[i][j]=0 and declare and initialize a third loop k from 0 to C1.

Step 14: find the product of the two matrix by multiplying A[i][k]\*B[k][i]) and add it in variable D[i][j].

Step 15: Print D[i]j] as the result of the product of two matrix

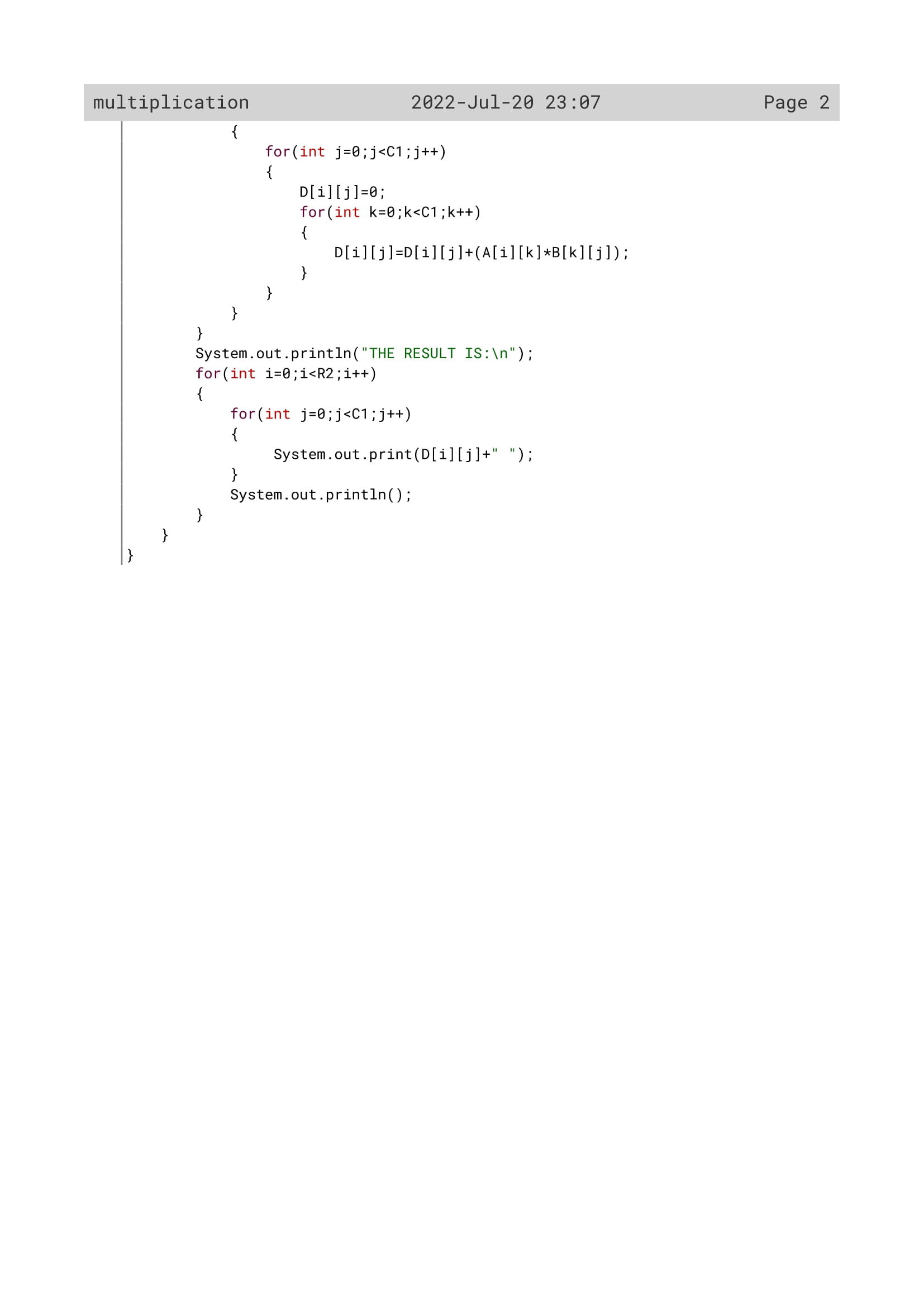
Step 16:Stop.

199

Source Code



200



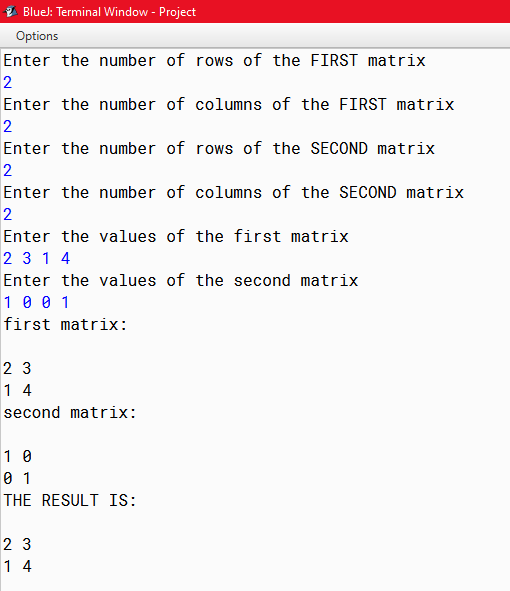
201

Variable Description

|  |  |  |
| --- | --- | --- |
| Variable name | Data type | Description |
| R1 | Int | To enter the number of rows of 1st matrix |
| C1 | Int | To enter the number of columns of 1st matrix |
| R2 | Int | To enter the number of rows of 2nd matrix |
| C2 | Int | To enter the number of columns of 2nd matrix |
| A[][] | int | To act as the 1st matrix |
| B[][] | Int | To act as the 2nd matrix |
| D[][] | Int | To act as resultant matrix |
| i | Int | To act as loop variable |
| j | Int | To act as loop variable |

202

OUTPUT



203