2D Array Problems -3

Assignment Solutions

Q1. Given a matrix arr[][] of integers, print the prefix sum matrix for it.

Input:
n = 3
m = 3
[[1,2,3],[4,5,6],[7,8,9]]

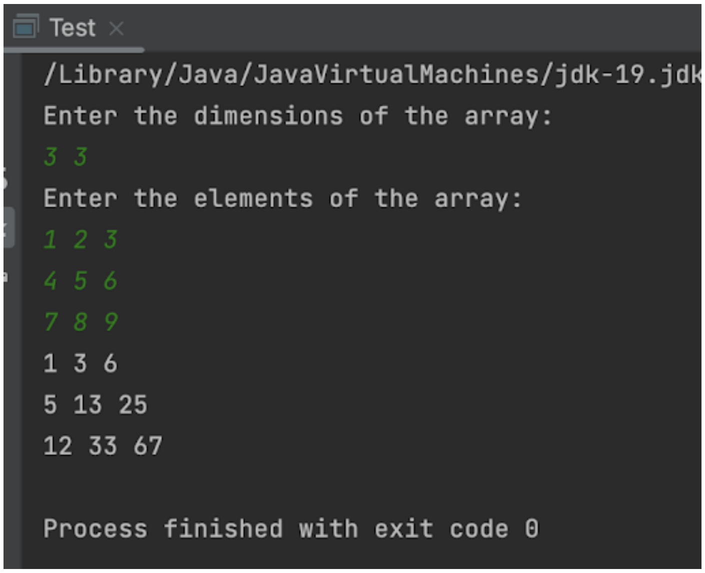
Expected Output:
[[1,3,6],[5,13,25],[12,33,67]]

Explanation:

- Prefix sum matrix for any cell can be calculated by the sum of its value, prefix sum of upper cell, prefix sum of left cell.
- For the first row and column, we calculate it by prefix sum of its value, and the cell before, for row the left cell, and for column, the upper cell.

Code:

```
import java.util.Scanner;
public class Test {
   public static void main(String[] args) {
       Scanner scn = new Scanner(System.in);
       System.out.println("Enter the dimensions of the array: ");
       int n = scn.nextInt();
       int m = scn.nextInt();
       int[][] mat = new int[n][m];
       System.out.println("Enter the elements of the array: ");
       for (int i = 0; i < n; i++) {
           for (int j = 0; j < m; j++) {
               mat[i][j] = scn.nextInt();
           }
       }
       int[][] arr = new int[n][m];
       for (int i = 0; i < n; i++) {
           for (int j = 0; j < m; j++) {
               arr[i][j] = mat[i][j];
               if(i == 0 && j == 0){
                   continue;
               }else if(i == 0){
                   arr[i][j] += arr[i][j-1];
               }else if(j == 0){
                   arr[i][j] += arr[i-1][j];
               }else{
                   arr[i][j] += arr[i-1][j] + arr[i][j-1];
               }
           }
       }
       for (int i = 0; i < n; i++) {
           for (int j = 0; j < m; j++) {
               System.out.print(arr[i][j] + " ");
           }
           System.out.println();
       }
   }
}
```



```
/Library/Java/JavaVirtualMachines/jdk-19.jdk
Enter the dimensions of the array:
3 3
Enter the elements of the array:
1 2 3
4 5 6
7 8 9
1 3 6
5 13 25
12 33 67

Process finished with exit code 0
```

Q2. A square matrix is said to be an perfect Matrix if both of the following conditions hold:
a) All the elements in the diagonals of the matrix are non-zero integers.
b) All other elements except the diagonal elements are 0.
Given a 2D integer array grid of size n*n representing a square matrix, return true if the grid is a perfect matrix. Otherwise, return false.

Input:
n = 4
arr[] = [[1,0,0,1],[0,2,1,0],[0,1,2,0],[3,0,0,1]]

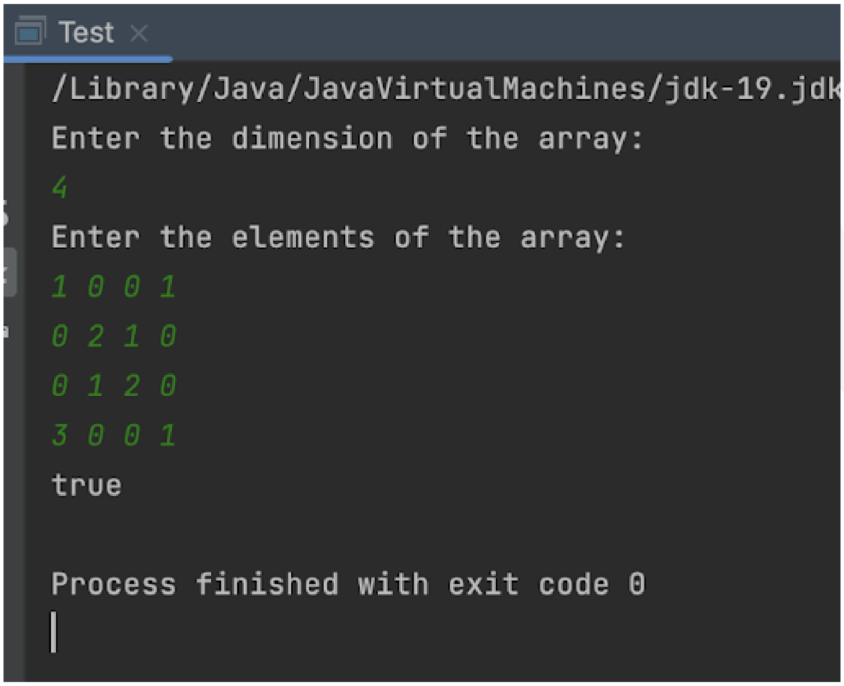
Expected Output:
true

Explanation:

- A matrix generally has 2 diagonals, which are formed by cells where row == column or row + column == n-1.

Code:

```
import java.util.Scanner;
public class Test {
   public static void main(String[] args) {
       Scanner scn = new Scanner(System.in);
       System.out.println("Enter the dimension of the array: ");
       int n = scn.nextInt();
       int[][] mat = new int[n][n];
       System.out.println("Enter the elements of the array: ");
       for (int i = 0; i < n; i++) {
           for (int j = 0; j < n; j++) {
               mat[i][j] = scn.nextInt();
           }
       }
       for (int i = 0; i < n; i++) {
           for (int j = 0; j < n; j++) {
               if(i == j || i+j == n-1){
                   if(mat[i][j] == 0){
                       System.out.println("false");
                       return;
                   }
}else{
                    if(mat[i][j] != 0){
                        System.out.println("false");
                        return;
                    }
                }
            }
        }
        System.out.println("true");
    }
}
```



```
/Library/Java/JavaVirtualMachines/jdk-19.jdk
Enter the dimension of the array:
4
Enter the elements of the array:
1 0 0 1
0 2 1 0
0 1 2 0
3 0 0 1
true

Process finished with exit code 0
```

Q3. Write a user defined function upper() which takes an integer square matrix as an input and its size N and prints the upper half of the matrix.

Input:
N=4
arr[][]=[[1,2,3,4],[5,6,7,8],[9,10,11,12],[13,14,15,16]]

Expected Output:

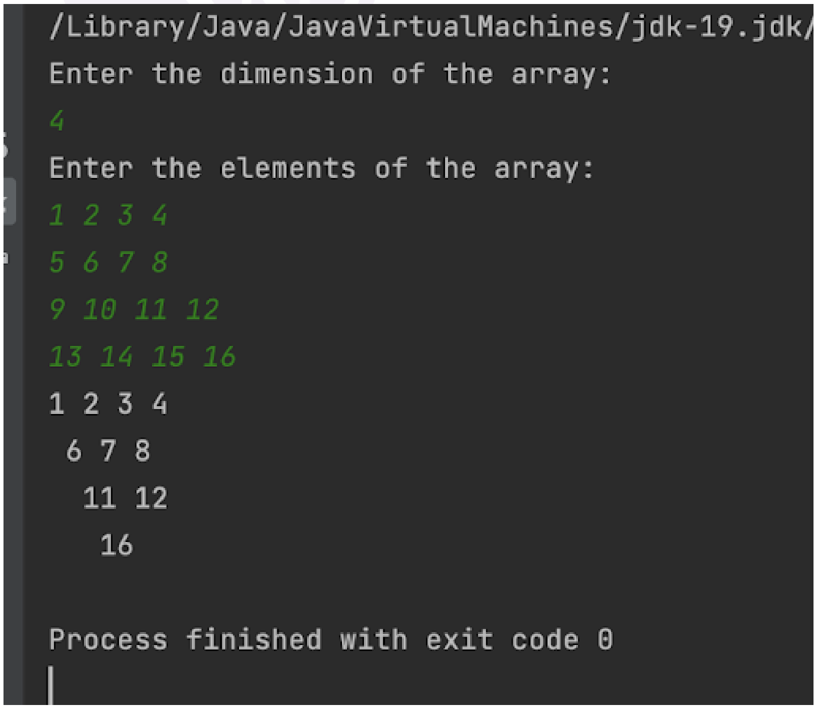
```
1 2 3 4
   6 7 8
     11 12
        16
```

Explanation:

- Elements of the upper matrix are those cells where column index exceeds or is equal to row index.

Code:

```
import java.util.Scanner;
public class Test {
   public static void main(String[] args) {
       Scanner scn = new Scanner(System.in);
       System.out.println("Enter the dimension of the array: ");
       int n = scn.nextInt();
       int[][] mat = new int[n][n];
       System.out.println("Enter the elements of the array: ");
       for (int i = 0; i < n; i++) {
           for (int j = 0; j < n; j++) {
               mat[i][j] = scn.nextInt();
           }
       }
       for (int i = 0; i < n; i++) {
           for (int j = 0; j < n; j++) {
               if(j >= i){
                   System.out.print(mat[i][j] + " ");
               }else{
                   System.out.print(" ");
               }
           }
           System.out.println();
       }
   }
}
```



```
/Library/Java/JavaVirtualMachines/jdk-19.jdk/
Enter the dimension of the array:
4
Enter the elements of the array:
1 2 3 4
5 6 7 8
9 10 11 12
13 14 15 16
1 2 3 4
 6 7 8
  11 12
   16

Process finished with exit code 0
```