1. Develop a JAVA program to add TWO matrices of suitable order N (The value of N should be read from command line arguments).

Save Filename as: MatrixAddition.java

Solution:-

```
public class MatrixAddition
{
    public static void main (String[] args)
  {
        int n = Integer.parseInt (args[0]);
       int[][] matrix1 = new int[n][n];
       int[][] matrix2 = new int[n][n];
      int[][] sum = new int[n][n];
     // Initialize matrices with some values, for example, i+j
      for (int i = 0; i < n; i++)
     {
        for (int j = 0; j < n; j++)
       {
            matrix1[i][j] = i + j;
            matrix2[i][j] = i + j;
      }
 }
 // Add the matrices
 for (int i = 0; i < n; i++)
 {
    for (int j = 0; j < n; j++)
    {
         sum[i][j] = matrix1[i][j] + matrix2[i][j];
 }
 }
 // Print the result
 System.out.println ("Sum of matrices is: ");
 for (int i = 0; i < n; i++)
 {
    for (int j = 0; j < n; j++)
   {
           System.out.print (sum[i][j] + " ");
  }
 System.out.println ();
}
}
}
```

Compile as: javacMatrixAddition.java

Run as:java MatrixAddition 3

Output:

```
Sum of matrices is:
0 2 4
2 4 6
6 8 7
```

2. Develop a stack class to hold a maximum of 10 integers with suitable methods. Develop a JAVA main method to illustrate Stack operations

Save Filename as:StackMain.java

Solution:-

```
import java.util.Scanner;

class Stack

{

    private int maxSize = 10;

    private int top;

    private int[] stackArray;

    public Stack ()

   {

       stackArray = new int[maxSize];

        top = -1;

   }

 public void push (int value)

 {

    if (top == maxSize - 1)

   {

       System.out.println("Stack is full. Unable to push " + value);

       return;

  }

     stackArray[++top] = value;

 }
public void pop ()
 {
    if (top == -1)
   {
          System.out.println ("Stack is empty");
          return;
    }
    System.out.println ("Popped " + stackArray[top--] + "from the stack");
}
 public void  display ()
 {
     if (top == -1)
    {
        System.out.println ("Stack is empty");
        return;
    }
    System.out.print ("Stack: ");
    for (int i = 0; i <= top; i++)
   {
      System.out.print (stackArray[i] + " ");
  }
   System.out.println ();
  }
}
```

```
public class StackMain

{

 public static void main (String[] args)

 {

     Stack stack = new Stack ();

     Scanner scanner = new Scanner (System.in);

     while (true)

     {

         System.out.println ("Choose an option:");

        System.out.println ("1) Push");

        System.out.println ("2) Pop");

        System.out.println ("3) Display");

         System.out.println ("4) Exit");

        int option = scanner.nextInt ();

       switch (option)

      {

         case 1:

                    System.out.println ("Enter a number to push:");

                     int num = scanner.nextInt ();

                     stack.push (num);

                      break;

 case 2:

             stack.pop ();

              break;
```

```
case 3:
    stack.display ();
    break;
case 4:
       scanner.close ();
       return;
 default:
              System.out.println("Invalid option.Please choose again.");
  }
}
}
}
```

Compile As: javacStackMain.java

Run As: java StackMain

Output:

```
Choose an option:
1) Push
2) Pop
3) Display
4) Exit
Enter a number to push:
10
Choose an option:
1) Push
2) Pop
```

```
3) Display
4) Exit
Enter a number to push:
20
Choose an option:
1) Push
2) Pop
3) Display
4) Exit
Enter a number to push:
30
Choose an option:
1) Push
2) Pop
3) Display
4) Exit
Stack: 10 20 30
Choose an option:
1) Push
2) Pop
3) Display
4) Exit
Popped 30 from the stack
```