Session I

Part I

Lab No. 1: Simple Java Programs using Control Structures

Lab Exercises

4 a. Write a method isPrime to accept one integer parameter and to check whether that parameter is prime or not.

Solution:

```
import java.util.Scanner;


public class PrimeCheck{
  public static void main(String args[])


  {


        Scanner sc = new Scanner(System.in);
        System.out.print("Enter The Number: ");
        int N = sc.nextInt();
         PrimeCheck obj = new PrimeCheck();
         obj.isPrime(N);
  }
   int isPrime(int x)
  {
     if(x==2||x%2!=0)
        {
        System.out.println("\nThe entered number is prime");
        }
     else
```

```
      {
          System.out.println("\nThe entered number is not prime");
      }
    return 0;
  }
}
```

Output:

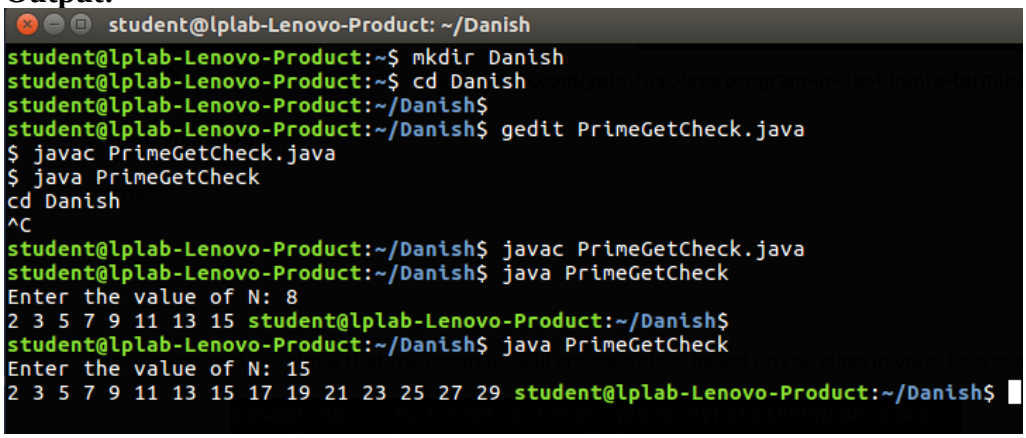
4 b. Using this method, generate first N prime numbers in the main method.

Solution:

```
import java.util.Scanner;
import java.lang.Math;
public class PrimeGetCheck
{
  public static int isPrime(int num)
          {
           if(num==2 || num%2!=0)
           return 1;
           else
           return 0;
          }
public static void main(String[] arg)
 {
  Scanner sc = new Scanner(System.in);
  System.out.print("Enter the value of N: ");
  int N = sc.nextInt();
```

```
  int counter=0, num=2;
  while(counter<N)
  {
   if(isPrime(num)==1)
    {
     System.out.print(num+" ");
     counter++;
    }
   num++;
  }
 }
}
```

Output:



Lab No. 2: 1D and 2D Arrays

Lab Exercises

1. Arrange the elements in ascending and descending order using Bubble sort method.

Solution:

```
import java.util.Scanner;
import java.lang.Math;

public class BubbleSort
{
    public static void main(String[] arg)
    {
        Scanner sc = new Scanner(System.in);
        System.out.println("Enter number of elements in the array: ");
        int n=sc.nextInt();
        int arr[] = new int[n];
        int temp;

            System.out.println("Enter all the elements of the array: ");
            for(int i=0; i<n; i++)
        {
         arr[i]=sc.nextInt();
        }

            System.out.println("\nThe sorted array in descending order: ");
        //Sorting the array in descending order
        for(int i=0; i<n; i++)
        {
            for(int j=1; j<n-i; j++)
            {
                if(arr[j]>arr[j-1])
```

```
        {
            temp=arr[j];
            arr[j]=arr[j-1];
            arr[j-1]=temp;
        }
    }
}

for(int i=0; i<n; i++)
{
    System.out.print(arr[i]+" ");
}

    System.out.println("\nThe sorted array in ascending order: ");
//Sorting the array in ascending order
for(int i=0; i<n; i++)
{
    for(int j=1; j<n-i; j++)
    {
        if(arr[j]<arr[j-1])
        {
            temp=arr[j];
            arr[j]=arr[j-1];
            arr[j-1]=temp;
        }
    }
}

for(int i=0; i<n; i++)
{
    System.out.print(arr[i]+" ");
```

```
        }
    }
}
```

Output:

```
@lplab-Lenovo-Product: ~/Danish
student@lplab-Lenovo-Product:~/Danish$ javac BubbleSort.java
student@lplab-Lenovo-Product:~/Danish$ java BubbleSort
Enter number of elements in the array:
5
Enter all the elements of the array:
254
2
3
45
0

The sorted array in descending order:
254 45 3 2 0
The sorted array in ascending order:
0 2 3 45 254 student@lplab-Lenovo-Product:~/Danish$ java BubbleSort
Enter number of elements in the array:
7
Enter all the elements of the array:
7 541 625 8 652 985 11002

The sorted array in descending order:
11002 985 652 625 541 8 7
The sorted array in ascending order:
7 8 541 625 652 985 11002 student@lplab-Lenovo-Product:~/Danish$
```

4. Find the addition of two matrices and display the resultant matrix.

Solution:

```
import java.util.Scanner;
import java.lang.Math;

public class AddArr
{
    public static void main(String arg[])
    {
        Scanner sc = new Scanner(System.in);
        System.out.print("Enter height of the matrices: ");
        int r=sc.nextInt();
        System.out.print("Enter width of the matrices: ");
        int c=sc.nextInt();
        int m1[][]=new int[r][c];
        int m2[][]=new int[r][c];
```

```
        System.out.println("Matrix 1: ");
        for(int i=0; i<r; i++)
        {
            for(int j=0; j<c; j++)
            {
                System.out.print("Enter value "+i+", "+j+": ");
                m1[i][j]=sc.nextInt();
            }
        }

        System.out.println("Matrix 2: ");
        for(int i=0; i<r; i++)
        {
            for(int j=0; j<c; j++)
            {
                System.out.print("Enter value "+i+", "+j+": ");
                m2[i][j]=sc.nextInt();
            }
        }
        System.out.print("\nThe resultant matrix is: \n");
            //Adding both the matrices
        for(int i=0; i<r; i++)
        {
            for(int j=0; j<c; j++)
            {
                m2[i][j]=m1[i][j]+m2[i][j];
                System.out.print(m2[i][j]+" ");
            }
            System.out.println();
        }
    }
}
```

Output:

```
@lplab-Lenovo-Product: ~/Danish
student@lplab-Lenovo-Product:~/Danish$ javac AddArr.java
student@lplab-Lenovo-Product:~/Danish$ java AddArr
Enter height of the matrices: 2
Enter width of the matrices: 2
Matrix 1:
Enter value 0, 0: 22
Enter value 0, 1: 341
Enter value 1, 0: 45
Enter value 1, 1: 6
Matrix 2:
Enter value 0, 0: 2
Enter value 0, 1: 44
Enter value 1, 0: 55
Enter value 1, 1: 112

The resultant matrix is:
24 385
100 118
student@lplab-Lenovo-Product:~/Danish$
```