

## LAB - 1

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

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

Code :

```
import java.util.Scanner;
```

```
class isPrime {
    //main function
    public static void main(String[] args) {
        int N, count = 0, k = 2;
        Scanner sc = new Scanner(System.in);
        System.out.println("Enter the limit : ");
        N = sc.nextInt();
        System.out.println("First " + N + " prime numbers are : ");
        while (count < N) {
            if (isPrime(k)) {
                System.out.print(k + " ");
                count++;
            }

            k++;
        }

    }

    //method
    public static boolean isPrime(int n) {
        int flag = 1;
        for (int i = 2; i <= Math.sqrt(n); i++) {
            if (n % i == 0) {
                flag = 0;
                break;
            }
        }
        if (flag == 1) {
            return (true);
        } else {
            return (false);
        }
    }
}
```

Output :

```
student@V310Z-000:~/Desktop/200905130/Lab 1$ pwd
/home/student/Desktop/200905130/Lab 1
student@V310Z-000:~/Desktop/200905130/Lab 1$ ls
isPrime.class  isPrime.java
student@V310Z-000:~/Desktop/200905130/Lab 1$ javac isPrime.java
student@V310Z-000:~/Desktop/200905130/Lab 1$ java isPrime
Enter the limit :
10
First 10 prime numbers are :
2 3 5 7 11 13 17 19 23 29 student@V310Z-000:~/Desktop/200905130/Lab 1$ java isPrime
Enter the limit :
3
First 3 prime numbers are :
2 3 5 student@V310Z-000:~/Desktop/200905130/Lab 1$ |
```

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

Code :

import java.util.Scanner;

```
class bubble_sort {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        int i, j, n;
        int [] a = new int[30];
        System.out.println("Enter the number of elements : ");
        n = sc.nextInt();

        System.out.println("Populate the array : ");
        for (i = 0; i < n; i++) {
            a[i] = sc.nextInt();
        }

        //ascending order
        for (i = 0; i < n - 1; i++) {
            for (j = 0; j < n - i - 1; j++) {
                if (a[j + 1] < a[j]) {
                    int temp = a[j + 1];
                    a[j + 1] = a[j];
                    a[j] = temp;
                }
            }
        }

        //printing ascending order
        System.out.print("\nThe ascending order of numbers : ");
        for (i = 0; i < n; i++) {
            System.out.print(a[i] + " ");
        }

        //descending order
        for (i = 0; i < n - 1; i++) {
            for (j = 0; j < n - i - 1; j++) {
                if (a[j + 1] > a[j]) {
                    int temp = a[j + 1];
                    a[j + 1] = a[j];
                    a[j] = temp;
                }
            }
        }
    }
}
```

```

    }
}

//printing descending order
System.out.print("\n\nThe descending order of numbers : ");
for (i = 0; i < n; i++) {
    System.out.print(a[i] + " ");
}
}
}

```

Ouput :

```

student@V310Z-000:~/Desktop/200905130/Lab 1$ pwd
/home/student/Desktop/200905130/Lab 1
student@V310Z-000:~/Desktop/200905130/Lab 1$ ls
1_1ss.png  bubble_sort.java  isPrime.class  isPrime.java
student@V310Z-000:~/Desktop/200905130/Lab 1$ javac bubble_sort.java
student@V310Z-000:~/Desktop/200905130/Lab 1$ java bubble_sort
Enter the number of elements :
10
Populate the array :
2
44
63
12
87
34
29
16
33
70

The ascending order of numbers : 2 12 16 29 33 34 44 63 70 87
The descending order of numbers : 87 70 63 44 34 33 29 16 12 2 student@V310Z-000:~/Desktop/200905130/Lab 1$ |

```

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

Code :

```
import java.util.Scanner;
```

```

class matrix_addition {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        int [][] a = new int [10][10],b = new int [10][10],c = new int [10][10];

        int m, n, p, q, i, j;
        System.out.println("Enter the dimensions of the 1st matrix : ");
        m = sc.nextInt();
        n = sc.nextInt();
        System.out.println("Enter the dimensions of the 2nd matrix : ");
        p = sc.nextInt();
        q = sc.nextInt();

        if ((m != p) || (n != q)) {
            System.out.println("\nThese two matrices cannot be added as they are of different
order.");
            System.exit(0);
        } else {

            System.out.println("\nEnter the elements of 1st matrix : ");

```

```

    for (i = 0; i < m; i++) {
        for (j = 0; j < n; j++) {
            a[i][j] = sc.nextInt();
        }
    }

    System.out.println("\nEnter the elements of 2nd matrix : ");
    for (i = 0; i < p; i++) {
        for (j = 0; j < q; j++) {
            b[i][j] = sc.nextInt();
        }
    }

    //Addition of two matrices
    for (i = 0; i < m; i++) {
        for (j = 0; j < n; j++) {
            c[i][j] = a[i][j] + b[i][j];
        }
    }

    //Printing the resultant matrix
    System.out.println("\nThe resultant matrix is : ");
    for (i = 0; i < m; i++) {
        for (j = 0; j < n; j++) {
            System.out.print(c[i][j] + " ");
        }
        System.out.println();
    }
}
}
}

```

Ouput :

```

student@V310Z-000:~/Desktop/200905130/Lab 1$ pwd
/home/student/Desktop/200905130/Lab 1
student@V310Z-000:~/Desktop/200905130/Lab 1$ ls
1_1ss.png 1_2ss.png bubble_sort.class bubble_sort.java isPrime.class isPrime.java matrix_addition.java
student@V310Z-000:~/Desktop/200905130/Lab 1$ javac matrix_addition.java
student@V310Z-000:~/Desktop/200905130/Lab 1$ java matrix_addition
Enter the dimensions of the 1st matrix :
2
2
Enter the dimensions of the 2nd matrix :
2
2
Enter the elements of 1st matrix :
1
2
3
4
Enter the elements of 2nd matrix :
4
3
2
1
The resultant matrix is :
5 5
5 5

```