

Lab-11

1. Write a program to implement Quick sort by selecting last element as pivot in each partition.

Code:

```
import java.util.*;

class Quicksort{

    static void swap(int[] a, int i, int j)
    {
        int temp = a[i];
        a[i] = a[j];
        a[j] = temp;
    }

    static int partition(int[] a, int low, int high)
    {
        int pivot = a[high];

        int i = (low - 1);

        for(int j = low; j <= high - 1; j++)
        {
            if (a[j] < pivot)
            {
```

```

        i++;
        swap(a, i, j);
    }
}
swap(a, i + 1, high);
return (i + 1);
}
static void quickSort(int[] a, int low, int high)
{
    if (low < high)
    {

        int pi = partition(a, low, high);
        quickSort(a, low, pi - 1);
        quickSort(a, pi + 1, high);
    }
}

static void printaay(int[] a, int size)
{
    for(int i = 0; i < size; i++)
        System.out.print(a[i] + " ");

    System.out.println();
}

```

```
public static void main(String[] args)
{
    Scanner sc = new Scanner(System.in);
    System.out.println("Enter the Number of elements to be in Array: ");
    int n = sc.nextInt();
    int[] a= new int[10];
    System.out.println("Enter the array elements: ");
    for(int i=0; i<n;i++)
    {
        a[i] =sc.nextInt();
    }
    System.out.println("Original array: ");
    printaay(a, n);
    quickSort(a, 0, n - 1);
    System.out.println("Sorted array: ");
    printaay(a, n);
}
}
```

```
D:\20BCE7323>javac Quicksort.java

D:\20BCE7323>java Quicksort
Enter the Number of elements to be in Array:
5
Enter the array elements:
86 24 68 7 34
Original array:
86 24 68 7 34
Sorted array:
7 24 34 68 86

D:\20BCE7323>
```

The screenshot displays a Java source file in an IDE and a corresponding Command Prompt window. The Java code implements a Quicksort algorithm. It includes a helper method `printaay` to print array elements and a `main` method that takes user input for the number of elements and the array values, then sorts them and prints the result.

```
38 }
39 }
40
41 static void printaay(int[] a, int size)
42 {
43     for(int i = 0; i < size; i++)
44         System.out.print(a[i] + " ");
45
46     System.out.println();
47 }
48
49 public static void main(String[] args)
50 {
51     Scanner sc = new Scanner(System.in);
52     System.out.println("Enter the Number of elements to be in Array: ");
53     int n = sc.nextInt();
54     int[] a = new int[10];
55     System.out.println("Enter the array elements: ");
56     for(int i=0; i<n;i++)
57     {
58         a[i] =sc.nextInt();
59     }
60     System.out.println("Original array: ");
61     printaay(a, n);
62     quickSort(a, 0, n - 1);
63     System.out.println("Sorted array: ");
64     printaay(a, n);
65 }
66 }
```

The Command Prompt window shows the execution of the program, matching the output seen in the first image:

```
D:\20BCE7323>javac Quicksort.java
D:\20BCE7323>java Quicksort
Enter the Number of elements to be in Array:
5
Enter the array elements:
86 24 68 7 34
Original array:
86 24 68 7 34
Sorted array:
7 24 34 68 86
D:\20BCE7323>
```

At the bottom of the IDE window, status information is displayed: "Java source file", "length: 1,323 lines: 66", "Ln: 55 Col: 54 Pos: 1,112", "Windows (CR LF)", "UTF-8", and "INS".