13 Jan 2021 Rinoy Kuriyakose R3 56

**Experiment: 18**

**Aim:**

|  |  |
| --- | --- |
|  |  |

Write a Java program that implements a Quick sort algorithm for sorting a list of names in ascending order.

**Concept Used:**

Quick Sort algorithm

**Algorithm:**

Algorithm Partition(A, p, r)

START

1. x = A[r]

2. i = p-1

3. for j = p to r

4. if (A[j] <= x)

5. i = i+1

6. if (i != j)

7. swap A[i] and A[j]

8. endif

9. endif

10. endfor

11. if (r != i+1)

12. swap A[i+1] and A[r]

13. endif

14. return i+1

STOP

Algorithm QuickSort(A, p, r)

START

1. if (p < r)

2. q = Partition(A, p ,r)

3. QuickSort(A, p, q-1)

4. QuickSort(A, q+1, r)

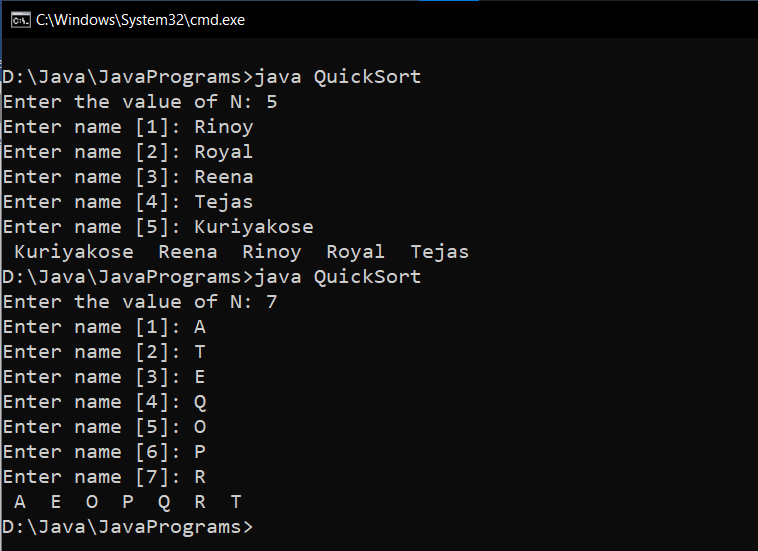
5. endif

STOP

**Program:**

import java.util.\*;  
public class QuickSort {  
 public static void main(String[] args) {  
 Scanner sc = new Scanner(System.in);  
 System.out.print("Enter the value of N: ");  
 int n = sc.nextInt();  
 String arr[] = new String[n];  
 sc.nextLine();  
 for(int i=0; i<n; i++)  
 {  
 System.out.print("Enter name [" + (i+1) +"]: ");  
 arr[i] = sc.nextLine();  
 }  
 QuickSort obj = new QuickSort();  
 obj.sort(arr, 0, arr.length - 1);  
 for (String i : arr) {  
 System.out.print(" "+ i + " ");  
 }  
 }  
 int partition(String [] arr, int p, int r) {  
 String pivot = arr[r];  
 int i = (p - 1);  
 String tmp;  
 for (int j = p ; j < r ; j++) {  
 if (arr[j].compareTo(pivot) < 0) {  
 i = i + 1;  
 tmp = arr[i];  
 arr[i] = arr[j];  
 arr[j] = tmp;  
 }  
 }  
 tmp = arr[i+1];  
 arr[i+1] = arr[r];  
 arr[r] = tmp;  
 return i + 1;  
 }  
 void sort(String[] arr, int p, int r) {  
 if (p < r) {  
 int q = partition(arr, p, r);  
 sort(arr, p, q - 1);  
 sort(arr, q + 1, r);  
 }  
 }  
}

**Output:**



**Result:**

Quick Sort is performed on a list of names.