Algorithm and Code(in Java)

1. Display name, student id and address of 5 students.

**Algorithm:**

1 Start the program

2 Define a 2-d string array(Students) which stores the value of student-id, student name and student-address

3 Initialize 2 variables int i and int j

4 repeat steps 4 to 8 until i<5

5 repeat steps 5 to 7 until j<3

6 print the value of Students[i][j]

7 print a space in the same line

8 go to next line

9 end

**Code:**

public class Practice  
{  
public static void main(String args[]){  
String [][] Students= {{"45", "A","UK"}, {"23", "B","USA"}, {"46", "C","INDIA"}, {"24", "D","KOREA"}, {"18", "E","AUSTRALIA"}};  
int i, j;  
for(i=0; i<5; i++){  
for(j=0; j<3; j++){  
System.out.print(Students[i][j] + " ");  
  
}  
System.out.println(" ");  
}  
}  
}

2 Sort the student id by using bubble sort

**Algorithm:**

1 Start the program

2 Declare an integer array of size 10, declare a temporary variable, int temp=0

3 Ask the user to input any 5 student id

4 Store the id into the array, array[]

4 repeat the steps 4 to 6 until i<5

5 if array[i]> array [i+1]

6 swap the value of array[i] and array[i+1]

7 display the value of array[]

8 end

**Code:**

import java.util.Scanner;public class Main\_Class { public static void main(String[] args) { int array[]=new int[10]; int temp=0; Scanner scan = new Scanner(System. in); for(int i=0;i<5;i++){ System.out.println("Enter your ID"); array[i]=scan.nextInt(); } System.out.println("Array before Applying bubble sort"); for(int i=0;i<5;i++) { System.out.println(+array[i]); } for(int i=0;i<5;i++) { for(int j =1;j<5-i;j++) { if(array[j]<array[j-1]) { temp=array[j-1];

array[j-1]=array[j];

array[j]=temp;

}

}

}

System.out.println("After Applying sorting: ");

for(int i=0;i<5;i++) {

System.out.println(+array[i]);

}

}

}

3 Sort the student id by using merge sort

**Algorithm:**

1 Start the program

2 make a function merge, which will divide the array into left and right arrays and merge the array after sorting

3 divide the array into L and R arrays from the middle of the original array

4 sort the L and R individually by dividing them into smaller arrays.

5 Repeat the division until each array has only one element

6 Combine and sort the array elements from left to right.

7 Once the array is fully combined and sorted, print it.

8 end the program

**Code:**

public class MergeSort

{ void merge(int arr[], int l, int m, int r)

{

int n1 = m - l + 1;

int n2 = r - m;

int L[] = new int[n1];

int R[] = new int[n2];

for (int i = 0; i < n1; ++i)

L[i] = arr[l + i];

for (int j = 0; j < n2; ++j)

R[j] = arr[m + 1 + j];

int i = 0, j = 0;

int k = l;

while (i < n1 && j < n2) {

if (L[i] <= R[j]) {

arr[k] = L[i];

i++;

}

else {

arr[k] = R[j];

j++;

}

k++;

}

while (i < n1) {

arr[k] = L[i];

i++;

k++;

}

while (j < n2) {

arr[k] = R[j];

j++;

k++;

}

}

void sort(int arr[], int l, int r)

{

if (l < r) {

int m =l+ (r-l)/2;

sort(arr, l, m);

sort(arr, m + 1, r);

merge(arr, l, m, r);

}

}

public static void main(String args[])

{

int arr[] = { 12, 11, 13, 5, 6, 7 };

System.out.println("Given Array");

for(int i=0; i<6; i++){

System.out.println(arr[i]);

}

MergeSort ob = new MergeSort();

ob.sort(arr, 0, arr.length - 1);

System.out.println("\nSorted array");

for(int i=0; i<6; i++){

System.out.println(arr[i]);

}

}

}