Algorithm 1

Step 1 : Start

Step 2 : Declare id as Integer Array , n , k as integer and name and address as 2D character array

Step3 : input n //number of students

Step 4 : initialize k=0

Step 5 : repeat step 6 to step 9 if k <n else goto step 11

Step 6 : input id[k]

Step 7 : input name[k]

Step 8 : input address[k]

Step 9 : k🡨k+1

Step 10 : goto step 5

Step11 : initialize k🡨0

Step 12 Repeat step 13 to 16 if k<n else goto step 18

Step 13 : print id[k]

Step 14 : print name[k]

Step 15 : print address[k]

Step 16 : k🡨k+1

Step17: goto step12

Step18 : stop

Algorithm 2

Step 1 : Start

Step 2 : Declare id as Integer Array , n , k,j as integer and name and address as 2D character array and tmp as character Array

Step3 : input n //number of students

Step 4 : initialize k🡨0

Step 5 : repeat step 6 to step 9 if k <n else goto step 11

Step 6 : input id[k]

Step 7 : input name[k]

Step 8 : input address[k]

Step 9 : k🡨k+1

Step 10 : goto step 5

Step11 : initialize k🡨0

Step 12 Repeat step 12 to 17 if k<n else goto step 18

Step 13 : print id[k] //printing before applying bubble sort

Step 14 : print name[k]

Step 15 : print address[k]

Step 16 : k🡨k+1

Step17: goto step10

Step18 : intializa i🡨0

Step 19 : Repeat step 18 to ifi<n else goto step else goto step 27

Step 20 : initialize j=0;

Step 21 : repeat step 22 to 24 if j🡨n-k else goto step 25

Step 22 : if id[j]>id[j+1]

Then

Temp 🡨id[j];

Id[j]🡨id[j+1];

Id[j+1]🡨temp;

Strcpy(tmp name[j];

Strcpy(name[j],name[j+1];

Strcpy(name[j+1],tmp);

Strcpy (name[j+1],tmp);

Strcpy(tmp,address[j]);

strcpy(address[j],address[j+1];

strcpy(address[j+1],tmp);

Step23 : j🡨j+1

Step24 : goto step21

Step25 : i🡨i+1

Step26 : gotgo step19

Step27: i🡨0

Step 28 : Repeat Step 29 to 33 if k<n else goto step34

Step29 : printid[k] //printing after applying bubble sort

Step 30 : print name[k]

Step31 : print address[k]

Step 32 : k🡨k+1

Step33 : goto step 28

Step 34 : Stop