10.a) Bubble sort is a sorting algorithm that works by repeatedly stepping through lists that need to be sorted, comparing each pair of adjacent items and swapping them if they are in the wrong order. This passing procedure is repeated until no swaps are required, indicating that the list is sorted. Bubble sort gets its name because smaller elements bubble toward the top of the list. Consider an array of size 10. It will be filled it by reading 10 integers. The final output will be sorted output in Ascending Order.

Program

#include <stdio.h> int main ()

{

int array[10], c, d, temp; printf("Enter the 10 elements\n"); for (c = 0; c < 10; c++)

{

scanf("%d", &array[c]);

}

for (c = 0 ; c < 9; c++)

{

for (d = 0 ; d < 9-c; d++)

{

if (array[d] > array[d+1])

{

temp = array[d]; array[d] = array[d+1];

array[d+1] = temp;

}

}

}

printf("Sorted elements\n");

for (c = 0; c < 10; c++)

{

printf("%d\n", array[c]);

}

return 0;

}

Sample input

Enter the 10 elements

3

5

6

7

8

9

11

13

45

32

Sorted elements

3

5

6

7

8

9

11

13

32 45

b) Insertion sort is a sorting algorithm in which the elements are transferred one at a time to the right position. Here the first element in the array is considered as sorted, even if it is an unsorted array. Then each element in the array is checked with the previous elements, resulting in a growing sorted output list. With each iteration, the sorting algorithm removes one element at a time and finds the appropriate location within the sorted array and inserts it there. The iteration continues until the whole list is sorted.

First an array of size 10 will be taken. We will fill it by reading 10 integers. The final output will be sorted output in Ascending Order.

#include<stdio.h> int main()

{

int i, j,key, a[25];

printf("Enter 10 elements:\n "); for(i=0;i<10;i++)

{

Scanf("%d",&a[i]);

}

for(i=1;i<10;i++)

{

key=a[i]; j=i-1; while((key<a[j])&&(j>=0))

{

a[j+1]=a[j]; j=j-1; } a[j+1]=key; }

printf(" Sorted elements:\n "); for(i=0;i<10;i++)

{

printf(" %d\n",a[i]);

}

return 0;

}

Sample input

Enter 10 elements:

3

5 4

2

1

7

8

13 34

22

Sorted elements:

1

2

3

4

5 7

8

13

22

34