#03. Write a program to sort a linear array using the bubble sort algorithm.

Theory: Bubble Sort is the simplest sorting algorithm that works by repeatedly swapping the adjacent elements if they are in the wrong order. The algorithm is not suitable for large data sets as its average and worst-case time complexity is quite high.

Suppose the list of numbers A[1], A[2]... ... A[N] is in memory. The bubble sort algorithm works as follows:

Step-1: Compare A[1] and A[2] and arrange them in the desired order, so that A[1] < A[2]. Then Compare A[2] and A[3]. Then compare A[2] < A[3]. Continue the process. So that A[N-1] < A[N].

Step-2: Repeat Step 1 less than N-1 comparation

Step-3: Repeat step 2 less than N-2 comparation.

For example: Sorted elements ascending order.

52, 32, 50, 40, 61
A_1 A_2 A_3 A_4 A_5

Pass 1. We have the following comparisons.

(a) Compare A_1 and A_2. Since 52 > 32 interchange 32 and 52 as follows:

(32), (52), 50, 40, 61

(b) Compare A_2 and A_3. 52 > 50 interchange as follows:

32, (50), (52), 40, 61

(c) Compare A_3 and A_4. 52 > 40 interchage as follows 32, 50, (40), (52), 61

(d) Compare A_4 and A_5. 52 < 61 not alternate.

32, 50, 40, (52), (61)

Pass 1 complete and 61 o.c's fixed.

Now pass 2 held with N-1 elements

Pass 2:

(32), (50), 40, 52, 61 not alternate.

32, (40), (50), 52, 61 interchange

32, 40, (50), (52), 61 not alternate.

Pass 3:

(32), (40), 50, 52 61 not alternate.

32, (40), (50), 52, 61 " "

Pass 4: (32), (40), 50, 52, 61 not alternate.

Finally the numbers are arranged in

ascending order.

Algorithm: (Bubble Sort) Bubble (Data, N)

Here Data is an array with N elements
This algorithm sorts the elements in
Data.

1. Repeat steps 2 and 3 for K=1 to N-1
2. Set PTR = 1.
3. Repeat while PTR $\leq$ (N-1-K):
   (a) If Data[PTR] > Data[PTR+1], then;
   Interchange Data[PTR] and Data[PTR+1
   (b) Set PTR = PTR+1.

4. Exit.

Source code: C++

```
#include<iostream>
using namespace std;
int main()
{
  int LA[5] = {5, 3, 6, 2, 7};
  int n = 5;
  int i, j;

  for(i=0; i<(n-1); i++)
  {
    j=0;
    while (j<(n-1-i))
    { if ( LA[j] > LA[j+1])
      {
        int temp = LA[j];
        LA[j] = LA[j+1];
        LA[j+1] = temp;
      }
    j=j+1;
    }
  }
  cout<< "The ascending array elements are"
      <<endl;

for(i=0; i<n; i++)
{
  cout<<"LA[i]"<<" ";
}
}
```

output:

The ascending array element are

2 3 5 6 7