氣泡排序法(Bubble Sort)是排序演算法的一種，其觀念是逐次比較相鄰的兩筆資料，然後根據排序條件交換(由小到大或由大到小)，一直交換到資料排序完成為止。

氣泡排序演算法的運作原理如下:

1.比較相鄰的元素，如果第一個元素比第二個元素大(小)，就交換這兩個元素。

2.對每一對相鄰的元素執行同樣的行為，從開始的第一對到結尾的最後一對。當這部做完之後最後的元素會是最大(小)的數。

3.針對除了最後一個已經排序過後的數之外重複步驟一跟步驟二。

4.持續每次對越來越少的未排序的元素重複上面的步驟，直到沒有任何一對數字需要比較。

執行時，未排序資料中的最大值會如同氣泡般往右邊移動，故此命名為氣泡排序法。

範例: 由小到大排序 3 7 1 2 9 4  
第一輪:  
3 7 1 2 9 4   
3 7 1 2 9 4  
3 1 7 2 9 4   
3 1 2 7 9 4  
3 1 2 7 9 4  
3 1 2 7 4 9  (此時9是最大的已排序的數，故第二輪無須交換他)  
第二輪:  
3 1 2 7 4 9  
1 3 2 7 4 9  
1 2 3 7 4 9  
1 2 3 7 4 9  
1 2 3 4 7 9 (此時7是第二大的已排序的數，故第三輪無須交換他)  
第三輪:  
1 2 3 4 7 9 (此時4是第三大的已排序的數，故第四輪無須交換他)  
第四輪:  
1 2 3 4 7 9 (此時3是第四大的已排序的數，故第五輪無須交換他)  
第五輪:  
1 2 3 4 7 9 (此時2是第五大的已排序的數，故第六輪無須交換他)  
第六輪:  
1 2 3 4 7 9 (此時1是第六大的已排序的數，且也已經沒有未排序的數，由此可知1 2 3 4 7 9是最終由小到大排序)

時間複雜度如下:

最佳時間複雜度：O(n):當資料的順序恰好是所選擇的排序方式時。

平均時間複雜度：O(n^2):第n筆資料，平均比較(n-1)/2 次。

最差時間複雜度：O(n^2):當資料的排序恰好是所選擇的相反排序方式時。

程式碼:

1. 由大到小---------------------------------------------------------------------------------------------------

#include <iostream>

using namespace std;

int main()

{

int number=0;

cin>>number;

int int\_array[number];

for(int int\_cnt=0;int\_cnt<number;int\_cnt++)

{

cin>>int\_array[int\_cnt];

}

for(int int\_cnt=0;int\_cnt<number;int\_cnt++)

{

for(int int\_cnt2=int\_cnt;int\_cnt2<number-1;int\_cnt2++)

{

if(int\_array[int\_cnt]<int\_array[int\_cnt2+1])

{

int int\_temp=int\_array[int\_cnt];

int\_array[int\_cnt]=int\_array[int\_cnt2+1];

int\_array[int\_cnt2+1]=int\_temp;

}

}

}

for(int int\_cnt=0;int\_cnt<5;int\_cnt++)

{

cout<<int\_array[int\_cnt]<<endl;

}

return 0;

}

1. 由小到大-----------------------------------------------------------------------------------------------

#include <iostream>

using namespace std;

int main()

{

int number=0;

cin>>number;

int int\_array[number];

for(int int\_cnt=0;int\_cnt<number;int\_cnt++)

{

cin>>int\_array[int\_cnt];

}

for(int int\_cnt=0;int\_cnt<number;int\_cnt++)

{

for(int int\_cnt2=int\_cnt;int\_cnt2<number-1;int\_cnt2++)

{

if(int\_array[int\_cnt]>int\_array[int\_cnt2+1])

{

int int\_temp=int\_array[int\_cnt];

int\_array[int\_cnt]=int\_array[int\_cnt2+1];

int\_array[int\_cnt2+1]=int\_temp;

}

}

}

for(int int\_cnt=0;int\_cnt<5;int\_cnt++)

{

cout<<int\_array[int\_cnt]<<endl;

}

return 0;

}