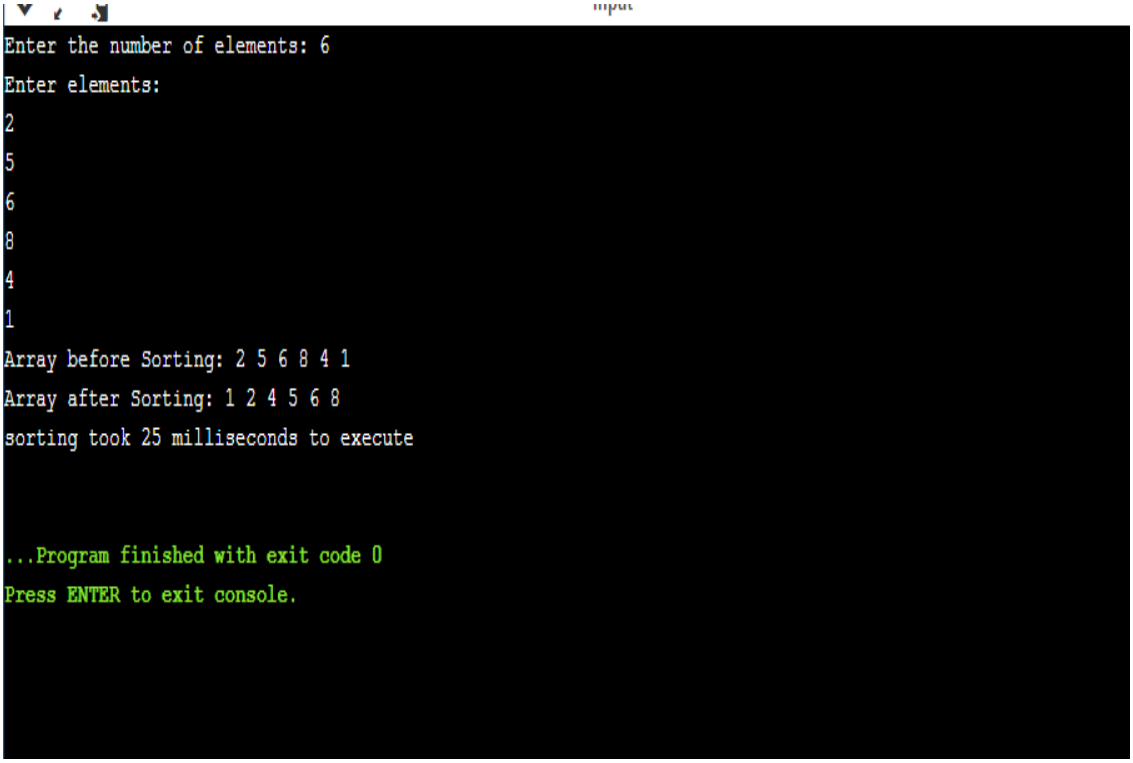
Bubblesort with time complexity:

```cpp
#include<iostream>
using namespace std;
void swapping(int &a, int &b) {      //swap the content of a and b
   int temp;
   temp = a;
   a = b;
   b = temp;
}
void display(int *array, int size) {
   for(int i = 0; i<size; i++)
      cout << array[i] << " ";
   cout << endl;
}
void bubbleSort(int *array, int size) {
   for(int i = 0; i<size; i++) {
      int swaps = 0;         //flag to detect any swap is there or not
      for(int j = 0; j<size-i-1; j++) {
         if(array[j] > array[j+1]) {       //when the current item is bigger than next
            swapping(array[j], array[j+1]);
            swaps = 1;    //set swap flag
         }
      }
      if(!swaps)
         break;       // No swap in this pass, so array is sorted
   }
}
```

```cpp
int main() {
   int n;
   clock_t t;
   cout << "Enter the number of elements: ";
   cin >> n;
   int arr[n];     //create an array with given number of elements
   cout << "Enter elements:" << endl;
   for(int i = 0; i<n; i++) {
      cin >> arr[i];
   }
   t=clock();
   cout << "Array before Sorting: ";
   display(arr, n);
   bubbleSort(arr, n);
   cout << "Array after Sorting: ";
   t=clock()-t;
   display(arr, n);
   double time_taken=1000000*((double)t)/CLOCKS_PER_SEC;
   cout<<"sorting took"<<" "<<time_taken<<" "<<"milliseconds to execute"<<endl;
}
```

Output:


Enter the number of elements: 6
Enter elements:
2
5
6
8
4
1
Array before Sorting: 2 5 6 8 4 1
Array after Sorting: 1 2 4 5 6 8
sorting took 25 milliseconds to execute
...Program finished with exit code 0
Press ENTER to exit console.