Seatwork 7.1	
Using Sorting Algorithms	
Course Code: CPE010	**Program:** Computer Engineering
Course Title: Data Structures and Algorithms	**Date Performed:** 9/16/25
Section: CPE21S4	**Date Submitted:** 9/16/25
Name(s): Avila, Vince Gabriel V.	**Instructor:** Engr. Jimlord Quejado
1. Output	

1. What is sorting algorithm?

A sorting algorithm is a way to put a list of things like numbers or words in order, such as from smallest to largest or A to Z. Its job is to quickly arrange the items in the list the way you want. These methods are very important in programming and are used in many different ways.

2.Where can sorting algorithms be used?

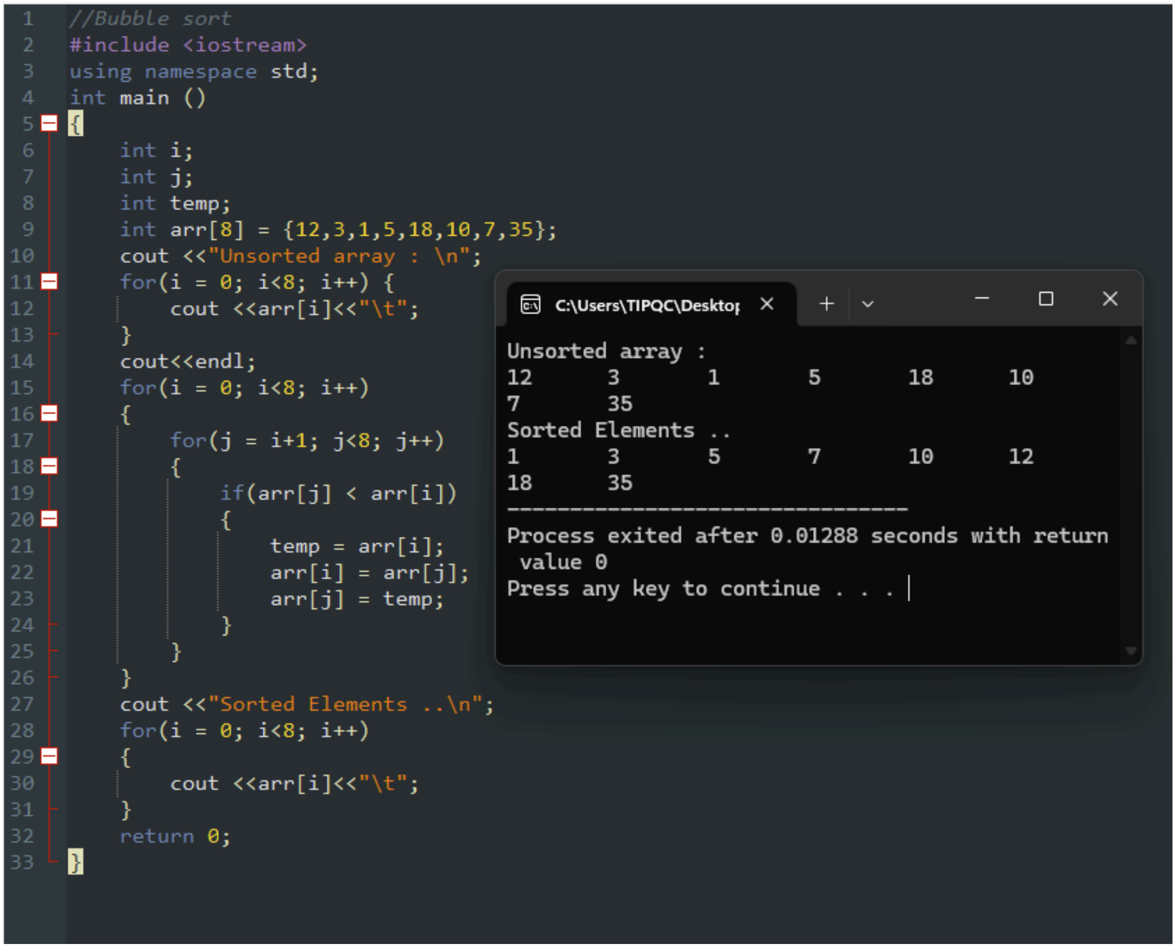
Sorting algorithms has wide applications in computer science and technology. They are very important for database management as they help manage data efficiently and organize records. Sorting is a pre-condition to doing a lot of search algorithms like binary search and similar which assume some sort of data ordering, if you want the list to be sorted faster in the first place. They are useful for data analysis, as well as task scheduling on a class of operating systems; and even computational geometry.

3. Explain the different types of sorting algorithms.

Sorting algorithms are basically ways to arrange data in a particular order such as from smallest to largest. Bubble Sort goes through pairs of elements and compares them, swapping if necessary until everything is sorted. Selection Sort is repeatedly getting to the smallest element and moving it in the beginning in Insertion Sort picks one element at a time and places it in the correct location among other sorted elements. Merge Sort breaks the list in smaller pieces, sorts and reassembles everything back into correct order. Each algorithm works in its way, and has value for different circumstances.

4. Give sample programs in C++ that uses sorting algorithms, specifically selection sort, insertion sort, and bubble sort. Explain how the programs work.

Bubble Sort:



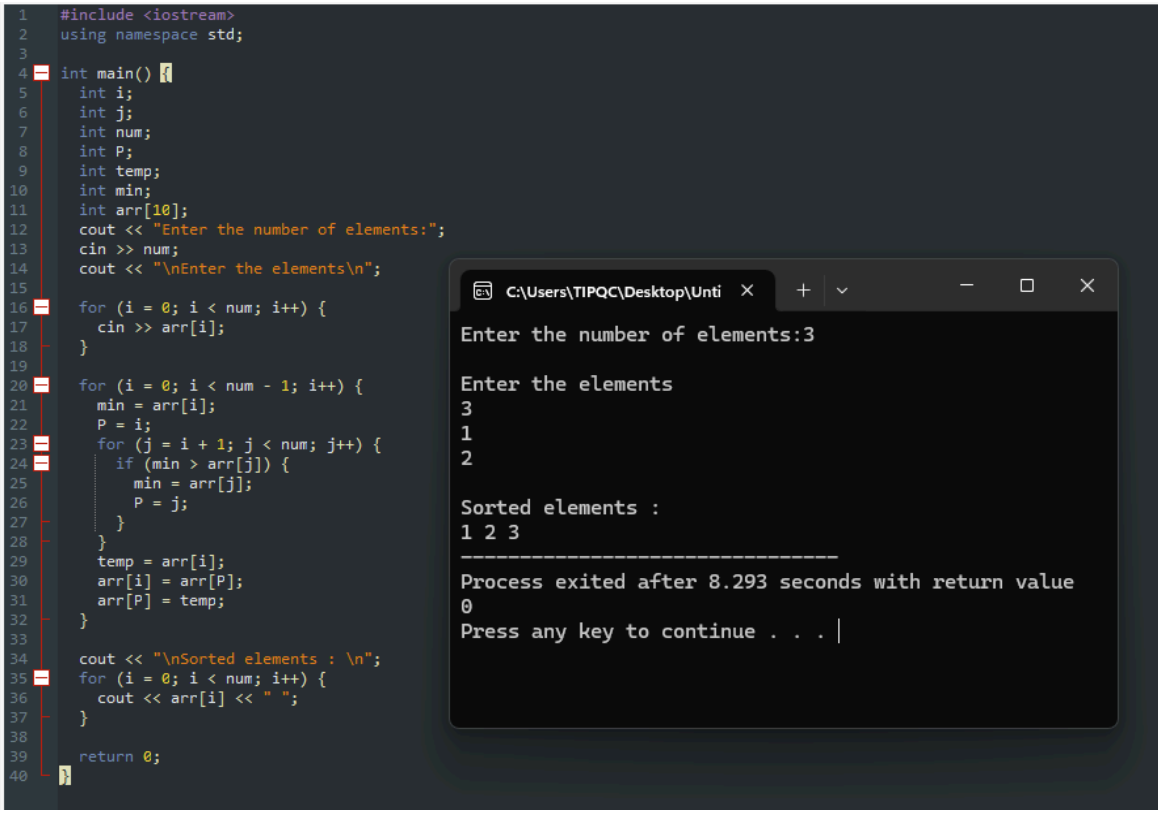
```cpp
//Bubble sort
#include <iostream>
using namespace std;
int main ()
{
    int i;
    int j;
    int temp;
    int arr[8] = {12,3,1,5,18,10,7,35};
    cout <<"Unsorted array : \n";
    for(i = 0; i<8; i++) {
        cout <<arr[i]<<"\t";
    }
    cout<<endl;
    for(i = 0; i<8; i++)
    {
        for(j = i+1; j<8; j++)
        {
            if(arr[j] < arr[i])
            {
                temp = arr[i];
                arr[i] = arr[j];
                arr[j] = temp;
            }
        }
    }
    cout <<"Sorted Elements ..\n";
    for(i = 0; i<8; i++)
    {
        cout <<arr[i]<<"\t";
    }
    return 0;
}
```

```
Unsorted array :
12      3       1       5       18      10
7       35
Sorted Elements ..
1       3       5       7       10      12
18      35
--------------------------------
Process exited after 0.01288 seconds with return
 value 0
Press any key to continue . . .
```

Analysis:

The bubble sort method to arrange an array of numbers from smallest to largest. It repeatedly compares and swaps adjacent numbers until the list is fully sorted.

Selection Sort:



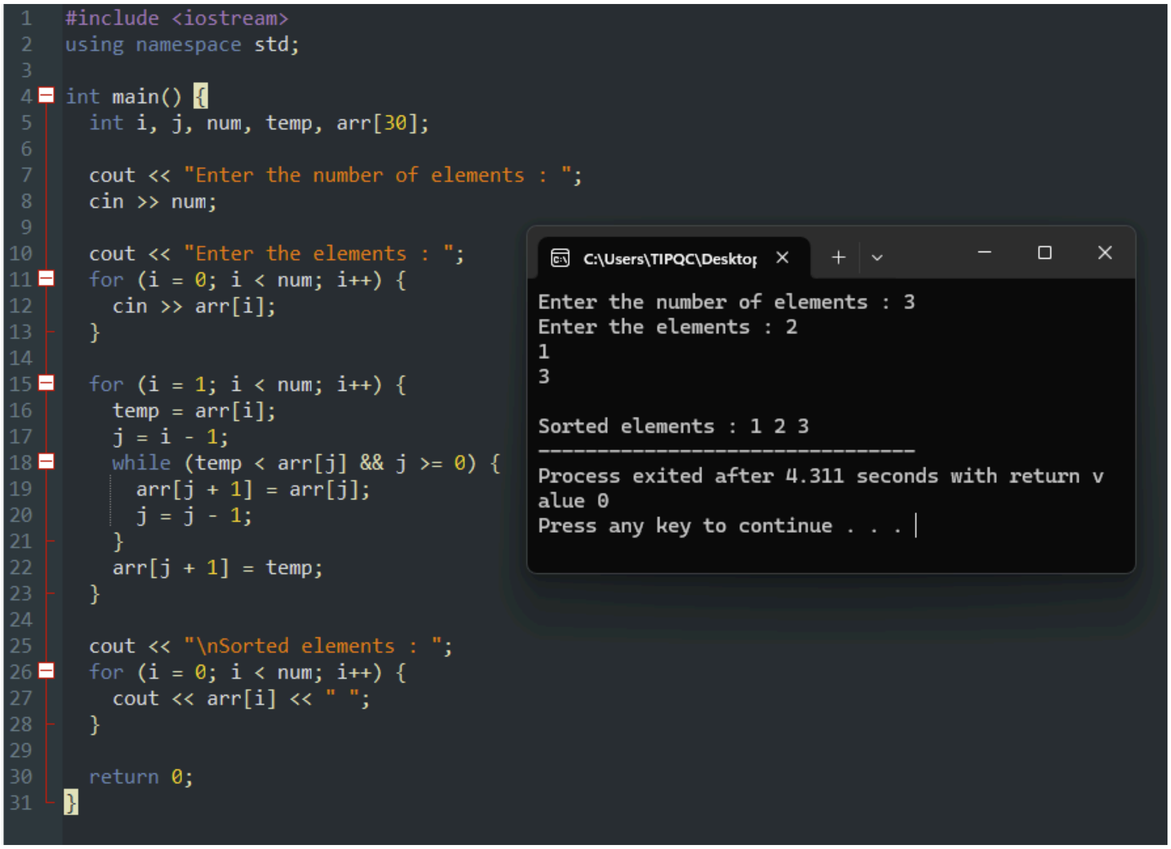
```
#include <iostream>
using namespace std;

int main() {
  int i;
  int j;
  int num;
  int P;
  int temp;
  int min;
  int arr[10];
  cout << "Enter the number of elements:";
  cin >> num;
  cout << "\nEnter the elements\n";

  for (i = 0; i < num; i++) {
    cin >> arr[i];
  }

  for (i = 0; i < num - 1; i++) {
    min = arr[i];
    P = i;
    for (j = i + 1; j < num; j++) {
      if (min > arr[j]) {
        min = arr[j];
        P = j;
      }
    }
    temp = arr[i];
    arr[i] = arr[P];
    arr[P] = temp;
  }

  cout << "\nSorted elements : \n";
  for (i = 0; i < num; i++) {
    cout << arr[i] << " ";
  }

  return 0;
}
```

```
Enter the number of elements:3

Enter the elements
3
1
2

Sorted elements :
1 2 3
--------------------------------
Process exited after 8.293 seconds with return value
0
Press any key to continue . . .
```

Analysis:

The Selection sort is to arrange numbers from smallest to largest by repeatedly finding the smallest remaining number and moving it to the front

Insertion Sort:



```
#include <iostream>
using namespace std;

int main() {
  int i, j, num, temp, arr[30];

  cout << "Enter the number of elements : ";
  cin >> num;

  cout << "Enter the elements : ";
  for (i = 0; i < num; i++) {
    cin >> arr[i];
  }

  for (i = 1; i < num; i++) {
    temp = arr[i];
    j = i - 1;
    while (temp < arr[j] && j >= 0) {
      arr[j + 1] = arr[j];
      j = j - 1;
    }
    arr[j + 1] = temp;
  }

  cout << "\nSorted elements : ";
  for (i = 0; i < num; i++) {
    cout << arr[i] << " ";
  }

  return 0;
}
```

```
Enter the number of elements : 3
Enter the elements : 2
1
3

Sorted elements : 1 2 3
--------------------------------
Process exited after 4.311 seconds with return v
alue 0
Press any key to continue . . .
```

Analysis:

The insertion sort is to arrange numbers from smallest to largest by picking a number and inserting it into its correct place in the sorted portion of the list.

APA Citation:

SS, R. A. (2025, January 26). *What is sorting in C++: bubble sort, insertion sort, merge sort & more*. Simplilearn.com.

https://www.simplilearn.com/tutorials/cpp-tutorial/sorting-in-cpp

2. Conclusion

In conclusion, Sorting algorithms are a set of algorithms that arrange data into a specified order such as numbers or words. Sorting algorithms are useful in technology because they are capable of taking large amounts of data and efficiently sorting that information making the user’s role of finding and organizing data simpler. It discuss about a number of sorting algorithms which include Bubble Sort, Insertion Sort, and Selection Sort but they use slightly different approaches to get to the same results.

3. Assessment Rubric