Section 3: Sorting

There are variety of ways to sort arrays using C++. Some come sort methods are bubble sort, selection sort, and quick sort. Listed below is a brief description of each sort.

Bubble Sort	This sort is the simplest way to sort a data set, but is very inefficient for large data structures. This sort works by using successive passes over the data set and moving the data in the desired order. The sort compares only elements that are adjacent to each other. The sweep is completed when there is no more comparisons to make from adjacent elements.
Selection Sort	Selection sort only sorts one element at a time. The user chooses one element at a time and compares it to every other element in the data set until it is the correct position. This is very inefficient way to sort since a large number of passes on the data set will need to be made in order to get the data set in the desired order.
Quick Sort	This sort is the quickest of sorts because uses the following strategies 1. The sort splits the data set in to subsets 2. The sort then sorts the sub sets 3. Finally the sort merges the sorted sub sets There are a number of ways to setup a quick sort; the key is as the programmer to pick a pivot point in which to begin sub creating sub sets (I.E choosing a point where all numbers on the left are less than the pivot point and all numbers to the right are larger than the pivot).
sort command line	The sort command line falls into the algorithm library. This option allows the user to send an arrays data set and sorts it. I.E my_array (first element, last element) = my_array (index, index+4) where index+4 is equal to the last item in the data set. This command helps to find the permutations to a given data set.

Section 3: Program 1: Bubble Sort

Directions: Type in the following Code

```
#include<iostream>
using namespace std;
int main()
{
      char a;
      int array[5], temp=0;                                    //declaring array
      cout<<"Enter 5 numbers randomly : "<<endl;
      for(int i=0; i<5; i++)
      {
       cin>>array[i];                                  //Taking input in array
      }
      cout<<endl;
      cout<<"Input array is: "<<endl;

      for(int j=0; j<5; j++)
      {
       cout<<"\t\t\tValue at "<<j<<" Index: "<<array[j]<<endl;      //Displaying Array
      }
      cout<<endl;
     for(int i2=0; i2<=4; i2++)                                    // Bubble Sort Starts Here
   {
     for(int j=0; j<4; j++)
     {
          if(array[j]>array[j+1])                                  //Swapping element in if statement
      {
       temp=array[j];
       array[j]=array[j+1];
       array[j+1]=temp;
      }
     }
   }
       cout<<"  Sorted Array is: "<<endl;                         // Displaying Sorted array
     for(int i3=0; i3<5; i3++)
   {
    cout<<"\t\t\tValue at "<<i3<<" Index: "<<array[i3]<<endl;
   }
cin >> a;
return 0;
}
```

Question

1. How is the variable temp being used in the sorting loop?

2. **Challenge:** Modify the code to make the program sort the data set in descending order. Show Teacher when completed.