Task# 1: Take an unsorted array of size 7. Sort the array in descending order using Quick Sort

technique. Take the first element as pivot element and dry run your code to find the right position

of that pivot element.

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



int it=1;
int partition(int arr[], int low, int high) {

   // Initialize pivot to be the first element
   int p = arr[low];
   int i = low;
   int j = high;
cout<<"pivot = "<<p<<endl;
   while (i < j) {

      // Find the first element greater than
      // the pivot (from starting)
      while (arr[i] <= p && i <= high - 1) {
         i++;
      }

      // Find the first element smaller than
      // the pivot (from last)
```

```
        while (arr[j] > p && j >= low + 1) {
            j--;
        }
        if (i < j) {
            swap(arr[i], arr[j]);
        }
    }
    swap(arr[low], arr[j]);


     cout<<"after "<<it<<" iteration ";
     for(int i=0;i<7;i++){

    cout<<arr[i]<<" ";
    }
                it++;
            cout<<endl;
    return j;

}

void quickSort(int arr[], int low, int high) {
    if (low < high) {

        // call partition function to find Partition Index
        int pi = partition(arr, low, high);

        // Recursively call quickSort() for left and right
        // half based on Partition Index
```

```
        quickSort(arr, low, pi - 1);
        quickSort(arr, pi + 1, high);
    }
}

int main() {

    int arr[7] = {1,4,10,7,3,5,2};
    int n = sizeof(arr) / sizeof(arr[0]);

    // calling quickSort() to sort the given array
    quickSort(arr, 0, n - 1);
            int arr2[7];
            for (int i=0;i<7;i++)
            {
                    arr2[i]=arr[6-i];
            }
    for (int i = 0; i < n; i++)
        cout<<" "<< arr2[i];

    return 0;
}
```

Task# 2: Take an unsorted array of size 5. Sort the array in descending order using Selection Sort technique. Dry run your code for each iteration. Implement the code in compiler and display the values of variables.

```
#include <iostream>
using namespace std;
// Function to perform Selection Sort
void SelectionSort(int arr[], int n)
{


}
int main() {
          int it=0;
  int n=5;
   int arr[n]={7,2,5,3,4};


    int i, j, min, temp;
   // Loop through each element in the array
   for (i = 0; i < n ; i++)
   {
      min = i;
      for (j = i ; j < n; j++)
      {
         if (arr[j] > arr[min])
         {
            min = j;
         }
      }
```

```
        cout<<it<<" iteration "<<endl;
     for(int o=0;o<5;o++){

        cout<<arr[o]<<" ";
    }
    cout<<endl;
    it++;
        temp = arr[min];
        arr[min] = arr[i];
        arr[i] = temp;
   }
    cout << "Sorted Array after Selection Sort: ";
    for (int i = 0; i < n; i++) {
        cout << arr[i] << " ";
    }
    cout << endl;
    return 0;
}
```