Course Name: A Level (2nd Sem) **Subject:** Data Struture using C++
Topic: Quick Sort in C++ **Date:** 17-04-2020

Example1 : Sorting an Array using Quick Sort.

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

void quickSort(int a[], int size);
void quickSort(int a[], int left, int right);
void choosePivot(int a[], int left, int right);
int partition(int a[], int left, int right);
void print(const int a[], int left, int right);

int main()
{
  // Test 1
  const int SIZE_1 = 8;
  int a1[SIZE_1] = {8, 4, 5, 3, 2, 9, 4, 1};

  print(a1, 0, SIZE_1 - 1);
  cout << endl;
  quickSort(a1, SIZE_1);
  print(a1, 0, SIZE_1 - 1);
  cout << endl << endl;

  // Test 2
  const int SIZE_2 = 13;
  int a2[SIZE_2] = {8, 4, 5, 3, 2, 9, 4, 1, 9, 1, 2, 4, 5};

  print(a2, 0, SIZE_2 - 1);
  cout << endl;
  quickSort(a2, SIZE_2);
  print(a2, 0, SIZE_2 - 1);
  cout << endl;
}

// Sort the given array of size
void quickSort(int a[], int size)
{
  quickSort(a, 0, size - 1);
}

// Sort the given array in [left, right]
void quickSort(int a[], int left, int right)
{
  if ((right - left) >= 1)
  {   // more than 1 elements, need to sort
    choosePivot(a, left, right);
    int pivotIndex = partition(a, left, right);
    quickSort(a, left, pivotIndex -  1);
    quickSort(a, pivotIndex + 1, right);
```

```
  }
}

// Choose a pivot element and swap with the right
void choosePivot(int a[], int left, int right)
{
  int pivotIndex = (right + left) / 2;
  int temp;
  temp = a[pivotIndex];
  a[pivotIndex] = a[right];
  a[right] = temp;
}

// Partition the array [left, right] with pivot initially on the right.
// Return the index of the pivot after partition, all elements to the
// left of pivot are smaller; while to the right are larger.
int partition(int a[], int left, int right)
{
  int pivot = a[right];
  int temp;  // for swapping
  int storeIndex = left;
    // Start the storeIndex from left, swap elements smaller than
    //  pivot into storeIndex and increase the storeIndex.
    // At the end of the pass, all elements up to storeIndex are
    //  smaller than pivot.
  for (int i = left; i < right; ++i)
 {  // exclude pivot
    if (a[i] < pivot)
   {
      // for tracing
      print(a, left, right);

      if (i != storeIndex)
      {
        temp = a[i];
        a[i] = a[storeIndex];
        a[storeIndex] = temp;
      }
      ++storeIndex;

      // for tracing
      cout << "=> ";
      print(a, left, right);
      cout << endl;
    }
  }
  // Swap pivot and storeIndex
  a[right] = a[storeIndex];
  a[storeIndex] = pivot;

  // for tracing
  print(a, left, storeIndex - 1);
  cout << "{" << a[storeIndex] << "} ";
  print(a, storeIndex + 1, right);
  cout << endl;
```

```
  return storeIndex;
}

// Print the contents of the given array from left to right (inclusive)
void print(const int a[], int left, int right)
{
  cout << "{";
  for (int i = left; i <= right; ++i)
  {
    cout << a[i];
    if (i < right) cout << ",";
  }
  cout << "} ";
}
```