CSE320 Lab Report .Cpp

Result for quicksort

```
Enter the size of the array: 5

 Enter the array:
        1 2 4 22 44

 Sorted Array is:
[006704029@csusb.edu@csevnc lan5Cpp]$ ./a.out
Enter the size of the array: 10

 Enter the array:
        12 32 45 11 34 54 67 62 67 99

 Sorted Array is:
        11      12      32      34      45      54      62      67      67
```

Result for Partition

```
[Running] cd "/home/csusb.edu/006704029/CSE320/lan5Cpp/
divided

[Done] exited with code=0 in 1.825 seconds
```

Source code

```
#include<iostream>
#include<vector>
using namespace std;
vector<int> arr;
void print_vector(){
    for(int x=0;x<arr.size();x++){
        cout<<"\t"<<arr[x];
    }
}
void swapp(int l, int t)
{
    int temp=arr[l];
    arr[l]=arr[t];
    arr[t]=temp;
}
int partition_array(int left,int right,int pivot)
{
    while(!(left==right && right==pivot && left==pivot))
    {
        while(arr[right]>=arr[pivot] && left<right)
        {
            right--;
        }
        swapp(right,pivot);
        pivot=right;
        while(arr[left]<=arr[pivot] && right>left)
        {
            left++;
        }
        swapp(left,pivot);
        pivot=left;
    }
    return pivot;
}
void quicksort(int left,int right,int pivot)
{
    if(left>=right)
        return;
    int index=partition_array(left,right,pivot);
    quicksort(left,index-1,left);
    quicksort(index+1,right,index+1);
}



int main()
{
    int s;
    cout<<"Enter the size of the array: ";
    cin>>s;
    cout<<"\n Enter the array:\n";
    int ele;
    for(int i=0; i<s; i++)
    {
        cout<<"\t";
        cin>>ele;
        arr.push_back(ele);
    }
    quicksort(0,s-1,0);
    cout<<"\n Sorted Array is:\n";
    print_vector();
    return 0;
}
```

```cpp
#include <bits/stdc++.h>
using namespace std;

bool isKPartitionPossibleRec(int arr[], int subsetSum[], bool taken[],
                  int subset, int K, int N, int curIdx, int limitIdx)
{
    if (subsetSum[curIdx] == subset)
    {

        if (curIdx == K - 2)
            return true;

        return isKPartitionPossibleRec(arr, subsetSum, taken, subset,
                                       K, N, curIdx + 1, N - 1);
    }

    for (int i = limitIdx; i >= 0; i--)
    {

        if (taken[i])
            continue;
        int tmp = subsetSum[curIdx] + arr[i];

        if (tmp <= subset)
        {

            taken[i] = true;
            subsetSum[curIdx] += arr[i];
            bool nxt = isKPartitionPossibleRec(arr, subsetSum, taken,
                                       subset, K, N, curIdx, i - 1);

            taken[i] = false;
            subsetSum[curIdx] -= arr[i];
            if (nxt)
                return true;
        }
    }
    return false;
}
```

```cpp
bool isKPartitionPossible(int arr[], int N, int K)
{

    if (K == 1)
        return true;

    if (N < K)
        return false;

    int sum = 0;
    for (int i = 0; i < N; i++)
        sum += arr[i];
    if (sum % K != 0)
        return false;

    int subset = sum / K;
    int subsetSum[K];
    bool taken[N];

    for (int i = 0; i < K; i++)
        subsetSum[i] = 0;

    for (int i = 0; i < N; i++)
        taken[i] = false;

    subsetSum[0] = arr[N - 1];
    taken[N - 1] = true;

    return isKPartitionPossibleRec(arr, subsetSum, taken,
                                   subset, K, N, 0, N - 1);
}

int main()
{
    int arr[] = {2, 1, 4, 5, 3, 3};
    int N = sizeof(arr) / sizeof(arr[0]);
    int K = 3;

    if (isKPartitionPossible(arr, N, K))
        cout <<"divided\n";
    else
        cout << "can't be divided.\n";
}
```

How easy/hard was it was to program?

Cpp is the language that I am most familiar with, it's not that easy as Python, but really easy to begin with.

The ease/difficulty of debugging:

Really easy to debug.

The speed of execution:

Run time already shown in the screenshot above