### Design and Analysis of Algorithms Assignment - 2

###### Name: Dhanraj Kore

###### Div: TY B

###### Roll No: 60

###### Batch : B-3

**Quick Sort Implementation**

**Approach 1: Simple Quick Sort:**

**CODE:**

#include <bits/stdc++.h>

using namespace std;

#define n 16

int Partition(int l, int h, int a[], int c[][n])

{

    int pivot = a[l];

    int i = l, j = h - 1;

    while (i < j)

    {

        while (a[i] <= pivot)

        {

            c[i][l]++;

            c[l][i]++;

            i++;

        }

        while (a[j] > pivot)

        {

            c[j][l]++;

            c[l][j]++;

            j--;

        }

        if (i < j)

            swap(a[i], a[j]);

    }

    swap(a[l], a[j]);

    return j;

}

void QuickSort(int a[], int c[][n], int l, int h)

{

    if (l < h)

    {

        int k = Partition(l, h, a, c);

        QuickSort(a, c, l, k);

        QuickSort(a, c, k + 1, h);

    }

}

int main()

{

    int a[n];

    int b[n];

    for (int i = 0; i < n; i++)

    {

        a[i] = rand() % 100 + 1;

    }

    cout << "\nThe Unsorted array is : " << endl

         << "\t\t";

    for (int i = 0; i < n; i++)

    {

        cout << a[i] << " ";

    }

    for (int i = 0; i < n; i++)

    {

        b[i] = a[i];

    }

    cout << endl;

    int c[n][n] = {0};

    QuickSort(a, c, 0, n);

    cout << "\nThe Sorted array is : " << endl

         << "\t\t";

    for (int i = 0; i < n; i++)

    {

        cout << a[i] << " ";

    }

    cout << endl;

    for (int i = 0; i < n; i++)

    {

        for (int j = i + 1; j < n; j++)

        {

            if (c[i][j] != 0)

            {

                cout << "\t(" << b[i] << "," << b[j] << ")=" << c[i][j] << endl;

            }

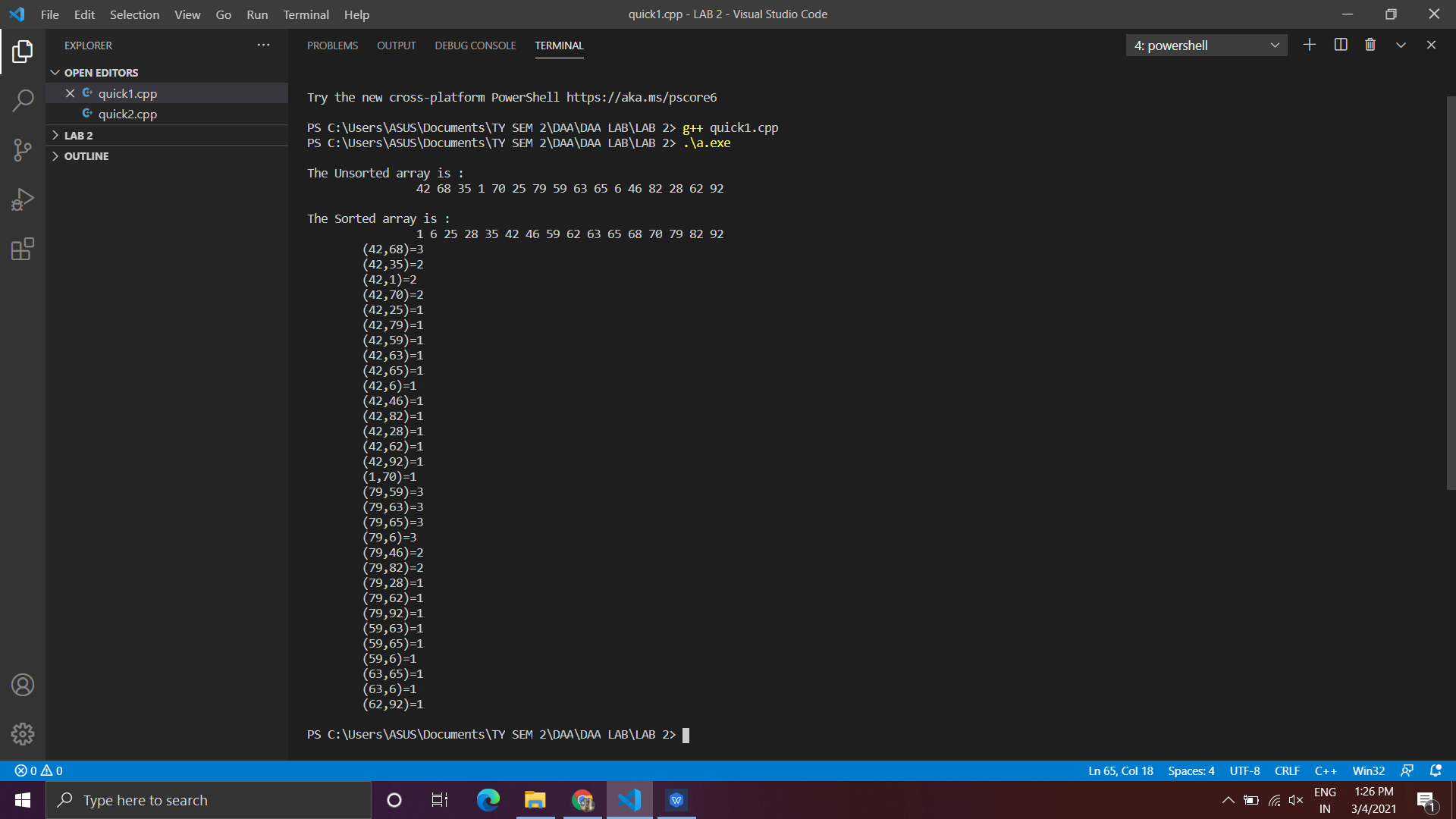
        }

    }

    cout << endl;

}

**O/P:**



**Approach 2: Randomized Quick Sort:**

**CODE:**

#include <bits/stdc++.h>

using namespace std;

#define n 15

int Partition(int l, int h, int a[], int c[][n])

{

    int idx = (rand() % (h - l)) + l;

    swap(a[l], a[idx]);

    int pivot = a[l];

    //cout<<a[idx]<<endl;

    int i = l, j = h - 1;

    while (i < j)

    {

        while (a[i] <= pivot)

        {

            c[i][l]++;

            c[l][i]++;

            i++;

        }

        while (a[j] > pivot)

        {

            c[j][l]++;

            c[l][j]++;

            j--;

        }

        if (i < j)

            swap(a[i], a[j]);

    }

    swap(a[l], a[j]);

    return j;

}

void QuickSort(int a[], int c[][n], int l, int h)

{

    if (l < h)

    {

        int k = Partition(l, h, a, c);

        QuickSort(a, c, l, k);

        QuickSort(a, c, k + 1, h);

    }

}

int main()

{

    int a[n];

    int b[n];

    for (int i = 0; i < n; i++)

    {

        a[i] = rand() % 100 + 1;

    }

    cout << "\nThe Unsorted array is : " << endl

         << "\t\t";

    for (int i = 0; i < n; i++)

    {

        cout << a[i] << " ";

    }

    for (int i = 0; i < n; i++)

    {

        b[i] = a[i];

    }

    cout << endl;

    int c[n][n] = {0};

    QuickSort(a, c, 0, n);

    cout << "\nThe Sorted array is : " << endl

         << "\t\t";

    for (int i = 0; i < n; i++)

    {

        cout << a[i] << " ";

    }

    cout << endl;

    for (int i = 0; i < n; i++)

    {

        for (int j = i + 1; j < n; j++)

        {

            if (c[i][j] != 0)

            {

                cout << "\t(" << b[i] << "," << b[j] << ")=" << c[i][j] << endl;

            }

        }

    }

    cout << endl;

}

**O/P:**

