

Experiment No: 01

Experiment Name: Sort and array using Quick Sort (Pivot = Highest Value)

Code:

```
#include <iostream>

using namespace std;

void swap(int& a, int& b) {                // Swap

    int temp = a;

    a = b;

    b = temp;

}

int partition(int arr[], int low, int high) {

    int pivot = arr[high]; // Set pivot to the highest value

    int i = low - 1;      // Next location of pivot

    for (int j = low; j < high; j++) {

        if (arr[j] <= pivot) {

            i++;

            swap(arr[i], arr[j]);

        }

    }

    swap(arr[i + 1], arr[high]);

    return (i + 1); // Return the partitioning index for Pivot

}

void quickSort(int arr[], int low, int high) {        // Quicksort function

    if (low < high) {

        int loc = partition(arr, low, high);
```

```

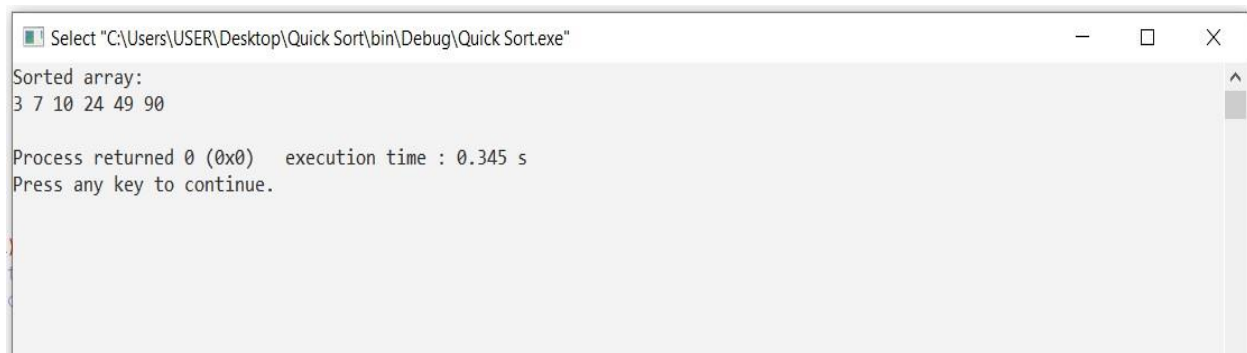
        // Sorting left and Right side of pivot
        quickSort(arr, low, loc - 1);
        quickSort(arr, loc + 1, high);
    }
}

void printArray(int arr[], int size) {           // Print the array function
    for (int i = 0; i < size; i++) {
        cout << arr[i] << " ";
    }
    cout << endl;
}

int main() {                                     // Main function
    int arr[] = {24, 49, 07, 90, 10, 3};
    int n = sizeof(arr) / sizeof(arr[0]);
    quickSort(arr, 0, n - 1);
    cout << "Sorted array: \n";
    printArray(arr, n);
    return 0;
}

```

Output:



```

Select "C:\Users\USER\Desktop\Quick Sort\bin\Debug\Quick Sort.exe"
Sorted array:
3 7 10 24 49 90
Process returned 0 (0x0)   execution time : 0.345 s
Press any key to continue.

```

