
Experiment 1.4

Student Name: Rajiv Paul

Branch: CSE

Semester: 3rd

Subject Name: Java Program Lab

UID: 20BCS1812

Section/Group: 6B

Date of Performance: 09/09/2021

Subject Code: 20CSP-219

Q1. Write a program to implement bubble sort. Given the numbers 7, 1, 4, 12, 67, 33, and 45. How many swaps will be performed to sort these numbers using the bubble sort ?

1) Aim/Overview of the practical:

To write a program to implement bubble sort.

2) Software required:

Vs Code

3) Source Code:

```
#include <iostream>
using namespace std;
int bubbleSort(int arr[], int n)
{
    int swapCount = 1, temp, count = 0;
    for (int pass = n - 1; pass >= 0 && swapCount; pass--)
    {
        swapCount = 0;
        for (int i = 0; i <= (pass - 1); i++)
        {
            if (arr[i] > arr[i + 1])
            {
                temp = arr[i];
                arr[i] = arr[i + 1];
                arr[i + 1] = temp;
                swapCount = 1;

                count++;
            }
        }
    }
    return count;
}
```

```
int main()
{
    int swaps;
    int arr[] = {7, 1, 4, 12, 67, 33, 45};
    int n = sizeof(arr) / sizeof(arr[0]);
    swaps = bubbleSort(arr, n);
    cout << "The array after sorting is: ";
    for (int i = 0; i < n; i++)
    {
        cout << arr[i] << " ";
    }
    cout << endl;
    cout << "The number of swaps is: " << swaps << endl;
    return 0;
}
```

4. Output:

```
The array after sorting is: 1 4 7 12 33 45 67  
The number of swaps is: 4
```

Q2. Write a program to sort an array of floating-point numbers in descending order using the quick sort ?

1) Aim/Overview of the practical:

To write a program to sort an array of floating-point numbers in descending order using the quick sort.

2) Software required:

Vs Code

3) Source Code:

```
#include <iostream>
using namespace std;
int partition(float a[], int low, int high)
{
    float pivoted_e = a[low];
    int left, right;
    left = low;
    right = high;
    while (left < right)
    {
        while (a[left] <= pivoted_e)
            left++;
        while (a[right] > pivoted_e)
            right--;
        if (left < right)
        {
            float temp = a[left];
            a[left] = a[right];
            a[right] = temp;
        }
    }
    a[low] = a[right];
    a[right] = pivoted_e;
    return right;
}
```

```
void quickSort(float arr[], int low, int high)
{
    int pivot;
    if (low < high)
    {
        pivot = partition(arr, low, high);
        quickSort(arr, low, pivot - 1);
        quickSort(arr, pivot + 1, high);
    }
}

void printArray(float arr[], int size)
{
    int i;
    for (i = 0; i < size; i++)
    {
        cout << arr[i] << " ";
    }
    cout << endl;
}
```

```
int main()
{
    float arr[] = { 9, 7, 6, 18, 32 };
    int n = sizeof(arr) / sizeof(arr[0]);
    quickSort(arr, 0, n - 1);
    cout << "The sorted array in descending order is: ";
    for (int i = 0; i <= n / 2; i++)
    {
        float temp = arr[i];
        arr[i] = arr[n - i - 1];
        arr[n - i - 1] = temp;
    }
    printArray(arr, n);
    return 0;
}
```

4. Output:

```
The sorted array in descending order is: 32 18 9 7 6
```