

Bubble Sort

```
File Edit Selection View Go Run Terminal Help Practical-2.cpp - DAA-practical - Visual Studio Code
Practical-2.cpp x C++ insertion-sort.cpp C++ selection-sort.cpp
Practical-2 > C++ Practical-2.cpp > ...
1 // Author: Anish Tilloo
2 // Roll No. : 34
3 // Program: Bubble Sort
4
5 #include <bits/stdc++.h>
6 using namespace std;
7
8 // Bubble Sort Algorithm
9 void bubbleSort(int n, int arr[]){
10     for (int i = 0; i < n; i++)
11     {
12         for (int j = 0; j < n - 1; j++)
13         {
14             if (arr[j] > arr[j + 1])
15             {
16                 // swapping if the number is greater than the next one
17                 swap(arr[j], arr[j + 1]);
18             }
19         }
20     }
21 }
22
23 void inputArray( int arr[], int arraySize){
24     cout << "Enter the elements of the array one by one " << endl;
25     for (int i = 0; i < arraySize; i++)
26     {
27         cin >> arr[i];
28     }
29 }
```

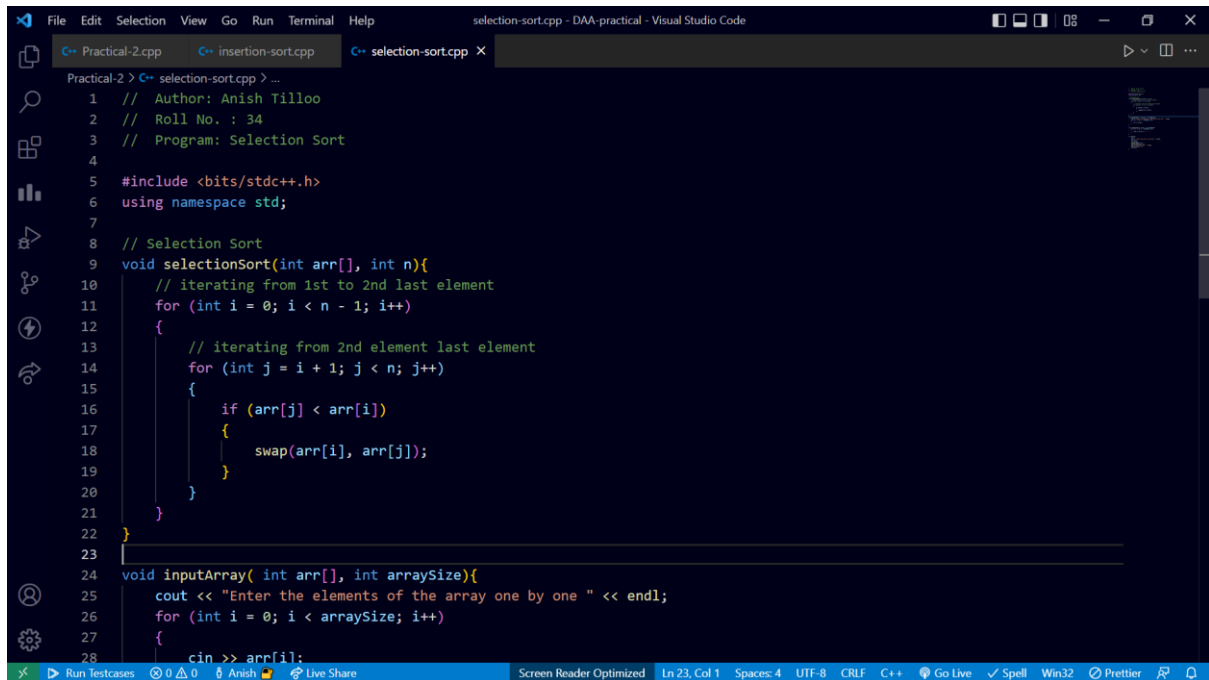
```
File Edit Selection View Go Run Terminal Help Practical-2.cpp - DAA-practical - Visual Studio Code
Practical-2.cpp x C++ insertion-sort.cpp C++ selection-sort.cpp
Practical-2 > C++ Practical-2.cpp > ...
23 void inputArray( int arr[], int arraySize){
24     cout << "Enter the elements of the array one by one " << endl;
25     for (int i = 0; i < arraySize; i++)
26     {
27         cin >> arr[i];
28     }
29 }
30
31 void printArray(int arr[], int arraySize){
32     for (int i = 0; i < arraySize; i++)
33     {
34         cout << arr[i] << " ";
35     }
36 }
37
38 int main(){
39     int n;
40     cout << "No. of elements in the array: " << endl;
41     cin >> n;
42     int arr[n];
43     inputArray(arr, n);
44     bubbleSort(n, arr);
45     cout << "Sorted Array: " << endl;
46     printArray(arr, n);
47     return 0;
48 }
```

The image shows a Visual Studio Code editor window with the title bar "Practical-2.cpp - DAA-practical - Visual Studio Code". The editor has three tabs open: "Practical-2.cpp", "insertion-sort.cpp", and "selection-sort.cpp". The "Practical-2.cpp" tab is active, showing a C++ program that uses `g++` to compile and run a sorting program. The program prompts the user for the number of elements in the array and then for the elements themselves. It then displays the sorted array.

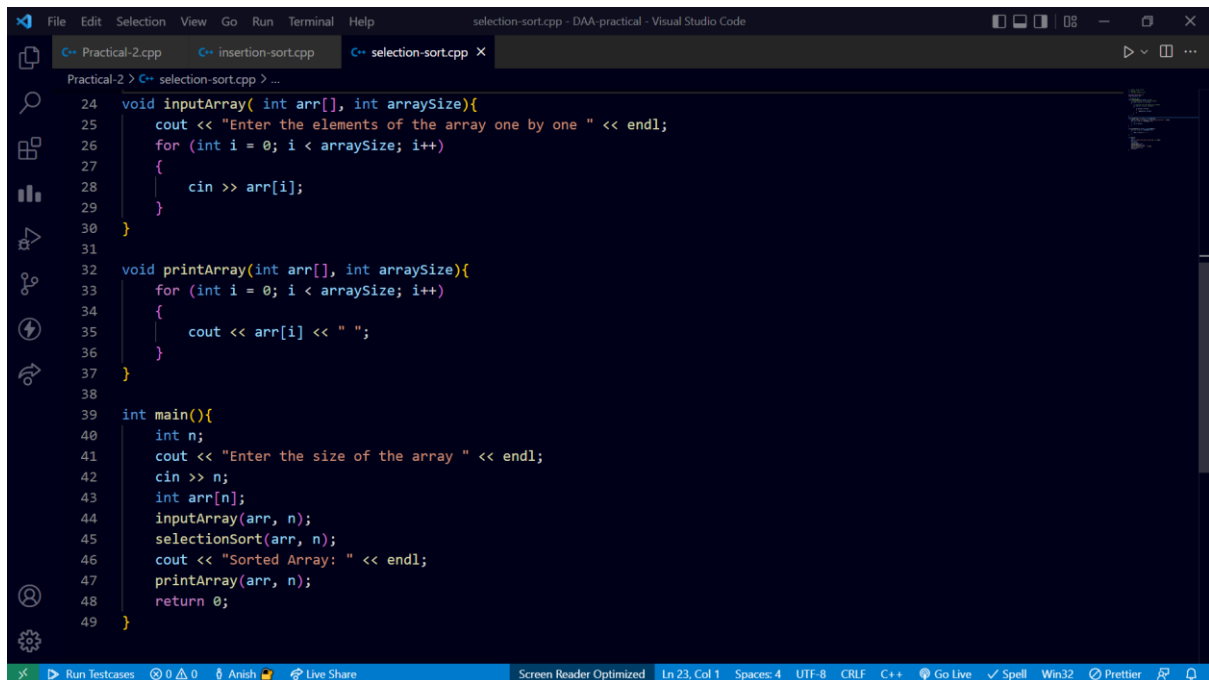
The terminal window at the bottom shows the execution of the program in a PowerShell environment. The user has navigated to the directory `E:\College-Work\Sixth-Semester\DAA-practical\Practical-2\` and run the program. The program prompts for the number of elements in the array (6) and then for the elements themselves (12 98 45 63 9 34). It then displays the sorted array (9 12 34 45 63 98).

```
Practical-2 > C++ Practical-2.cpp > ...  
Copyright (c) Microsoft Corporation. All rights reserved.  
  
Try the new cross-platform PowerShell https://aka.ms/pscore6  
  
PS E:\College-Work\Sixth-Semester\DAA-practical\Practical-2> cd "E:\College-Work\Sixth-Semester\DAA-practical\Practical-2\" ; if ($?) { g++ Pract  
ical-2.cpp -o Practical-2 } ; if ($?) { .\Practical-2 }  
No. of elements in the array:  
6  
Enter the elements of the array one by one  
12 98 45 63 9 34  
Sorted Array:  
9 12 34 45 63 98  
PS E:\College-Work\Sixth-Semester\DAA-practical\Practical-2> cd "E:\College-Work\Sixth-Semester\DAA-practical\Practical-2\" ; if ($?)  
{ g++ Practical-2.cpp -o Practical-2 } ; if ($?) { .\Practical-2 }  
No. of elements in the array:  
5  
Enter the elements of the array one by one  
5 4 3 2 1  
Sorted Array:  
1 2 3 4 5  
PS E:\College-Work\Sixth-Semester\DAA-practical\Practical-2> |
```

Selection Sort



```
1 // Author: Anish Tilloo
2 // Roll No. : 34
3 // Program: Selection Sort
4
5 #include <bits/stdc++.h>
6 using namespace std;
7
8 // Selection Sort
9 void selectionSort(int arr[], int n){
10     // iterating from 1st to 2nd last element
11     for (int i = 0; i < n - 1; i++)
12     {
13         // iterating from 2nd element last element
14         for (int j = i + 1; j < n; j++)
15         {
16             if (arr[j] < arr[i])
17             {
18                 swap(arr[i], arr[j]);
19             }
20         }
21     }
22 }
23
24 void inputArray( int arr[], int arraySize){
25     cout << "Enter the elements of the array one by one " << endl;
26     for (int i = 0; i < arraySize; i++)
27     {
28         cin >> arr[i];
29     }
30 }
```



```
24 void inputArray( int arr[], int arraySize){
25     cout << "Enter the elements of the array one by one " << endl;
26     for (int i = 0; i < arraySize; i++)
27     {
28         cin >> arr[i];
29     }
30 }
31
32 void printArray(int arr[], int arraySize){
33     for (int i = 0; i < arraySize; i++)
34     {
35         cout << arr[i] << " ";
36     }
37 }
38
39 int main(){
40     int n;
41     cout << "Enter the size of the array " << endl;
42     cin >> n;
43     int arr[n];
44     inputArray(arr, n);
45     selectionSort(arr, n);
46     cout << "Sorted Array: " << endl;
47     printArray(arr, n);
48     return 0;
49 }
```

The image shows a Visual Studio Code window with the file `selection-sort.cpp` open. The code defines a `main` function that takes an integer `n` and prompts the user to enter the size of the array and its elements. It then uses a selection sort algorithm to sort the array and prints the sorted result.

```
38
39 int main(){
40     int n;
```

The terminal output shows two successful runs of the program. In the first run, the user enters 6 and the array [87, 65, 99, 107, 27, 5], resulting in the sorted array [5, 27, 65, 87, 99, 107]. In the second run, the user enters 8 and the array [9, 8, 19, 1, 0, 24, 56, 99], resulting in the sorted array [0, 1, 8, 9, 19, 24, 56, 99].

```
PS E:\College-Work\Sixth-Semester\DAA-practical\Practical-2> cd "e:\College-Work\Sixth-Semester\DAA-practical\Practical-2\" ; if ($?)
{ g++ selection-sort.cpp -o selection-sort } ; if ($?) { .\selection-sort }
Enter the size of the array
6
Enter the elements of the array one by one
87 65 99 107 27 5
Sorted Array:
5 27 65 87 99 107
PS E:\College-Work\Sixth-Semester\DAA-practical\Practical-2> cd "e:\College-Work\Sixth-Semester\DAA-practical\Practical-2\" ; if ($?)
{ g++ selection-sort.cpp -o selection-sort } ; if ($?) { .\selection-sort }
Enter the size of the array
8
Enter the elements of the array one by one
9 8 19 1 0 24 56 99
Sorted Array:
0 1 8 9 19 24 56 99
PS E:\College-Work\Sixth-Semester\DAA-practical\Practical-2> |
```

Insertion Sort

```
File Edit Selection View Go Run Terminal Help insertion-sort.cpp - DAA-practical - Visual Studio Code
C++ Practical-2.cpp C++ insertion-sort.cpp X C++ selection-sort.cpp
Practical-2 > C++ insertion-sort.cpp > ...
1 // Author: Anish Tilloo
2 // Roll No. : 34
3 // Program: Insertion Sort
4
5 #include <bits/stdc++.h>
6 using namespace std;
7
8 void insertionSort(int arr[], int n){
9     int key, j;
10    // here we are assuming the first element is already sorted so we are starting the for loop with 1
11    for (int i = 1; i < n ; i++)
12    {
13        key = arr[i];
14        j = i - 1;
15        // here we are checking if the current element is greater the key
16        // and also checking j is greater than or equal to 0
17        // if the above conditions are true then we are incrementing the index og the current element
18        while (arr[j] > key && j >= 0)
19        {
20            arr[j + 1] = arr[j];
21            j--;
22        }
23        arr[j + 1] = key;
24    }
25 }
26
27 void inputArray( int arr[], int arraySize){
28     cout << "Enter the elements of the array one by one " << endl;
```

```
File Edit Selection View Go Run Terminal Help insertion-sort.cpp - DAA-practical - Visual Studio Code
C++ Practical-2.cpp C++ insertion-sort.cpp X C++ selection-sort.cpp
Practical-2 > C++ insertion-sort.cpp > ...
29     for (int i = 0; i < arraySize; i++)
30     {
31         cin >> arr[i];
32     }
33 }
34
35 void printArray(int arr[], int arraySize){
36     for (int i = 0; i < arraySize; i++)
37     {
38         cout << arr[i] << " ";
39     }
40 }
41
42 int main(){
43     int n;
44     cout << "Enter the size of the array " << endl;
45     cin >> n;
46     int arr[n];
47     inputArray(arr, n);
48     insertionSort(arr, n);
49     cout << "Sorted Array" << endl;
50     printArray(arr, n);
51     return 0;
52 }
```

```
{ g++ insertion-sort.cpp -o insertion-sort } ; if ($?) { .\insertion-sort }\Practical-2> cd "e:\College-Work\Sixth-Semester\DAA-practical\Practical-2"
Enter the size of the array
6
Enter the elements of the array one by one
108 67 1 109 2 9
Sorted Array
1 2 9 67 108 109
PS E:\College-Work\Sixth-Semester\DAA-practical\Practical-2> cd "e:\College-Work\Sixth-Semester\DAA-practical\Practical-2\" ; if ($?) { g++ insertion-sort.cpp -o insertion-sort } ; if ($?) { .\insertion-sort }
Enter the size of the array
5
Enter the elements of the array one by one
98 76 54 32 100
Sorted Array
32 54 76 98 100
PS E:\College-Work\Sixth-Semester\DAA-practical\Practical-2>
```