

# Sorting

## 1. Insertion Sorting:

### Code:

```
#include <iostream>

using namespace std;

void Insertion_Sort(int Arr[], int size )
{
    for (int i = 1; i < size; i++)
    {
        int key = Arr[i];
        int j = i - 1;
        while (j >= 0 && Arr[j] > key)
        {
            Arr[j + 1] = Arr[j];
            j = j - 1;
        }
        Arr[j + 1] = key;
    }
}

int main()
{
    int size;
    cout << "Enter the size of Array: ";
    cin >> size;
    int Arr[size];
    cout << "Please Enter the values for filling the indexes of the array" << endl;
    for (int i = 0; i < size; i++)
    {
```

```

        cout << "Enter the value for index " << i << ": ";

        cin >> Arr[i];
    }

    cout << "Unsorted Array" << endl;

    for (int i = 0; i < size; i++)
    {
        cout << Arr[i] << " ";
    }

    Insertion_Sort(Arr, size);

    cout << "\nSorted Array" << endl;

    for (int i = 0; i < size; i++)
    {
        cout << Arr[i] << " ";
    }

    return 0;
}

```

## Output:

The screenshot shows the Dev-C++ IDE with the 'Insertion Sort.cpp' file open. The console window displays the following output:

```

Enter the size of Array: 4
Please Enter the values for filling the indexes of the array
Enter the value for index 0: 1
Enter the value for index 1: 7
Enter the value for index 2: 4
Enter the value for index 3: 3
Unsorted Array
1 7 4 3
Sorted Array
1 3 4 7
-----
Process exited after 9.587 seconds with return value 0
Press any key to continue . . .

```

The IDE interface includes a menu bar (File, Edit, Search, View, Project, Execute, Tools, AStyle, Window, Help), a toolbar, and a project explorer on the left. The status bar at the bottom shows the current line (29), column (38), and other details.

## 2.Selection Sorting:

### Code:

```
#include <iostream>

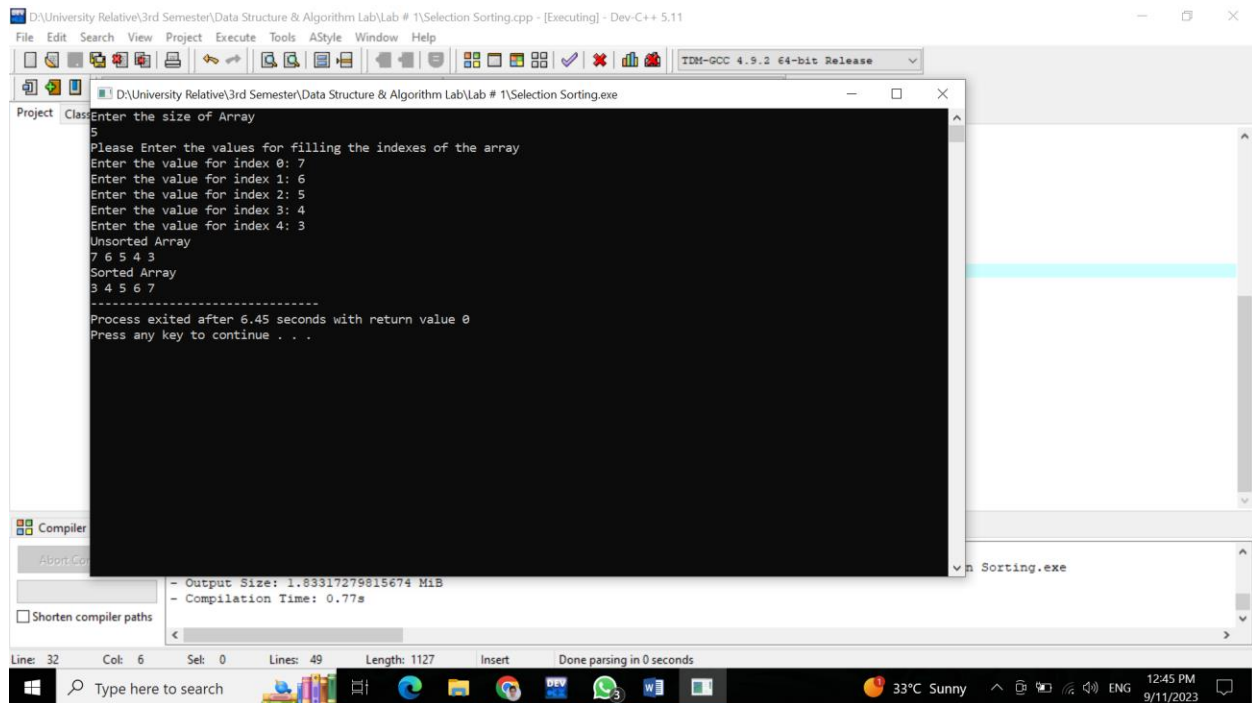
using namespace std;

void Selection_Sort(int Arr[], int size )
{
    for (int i = 0; i < size - 1; i++)
    {
        int small = i;
        for (int j = i + 1; j < size; j++)
        {
            if (Arr[j] < Arr[small])
            {
                small = j;
            }
        }
        if (small != i)
        {
            int temp;
            temp = Arr[small];
            Arr[small] = Arr[i];
            Arr[i] = temp;
        }
    }
}

int main()
{
    int size;
```

```
cout << "Enter the size of Array" << endl;
cin >> size;
int Arr[size];
cout << "Please Enter the values for filling the indexes of the array" << endl;
for (int i = 0; i < size; i++)
{
    cout << "Enter the value for index " << i << ": ";
    cin >> Arr[i];
}
cout << "Unsorted Array" << endl;
for (int i = 0; i < size; i++)
{
    cout << Arr[i] << " ";
}
Selection_Sort(Arr, size);
cout << "\nSorted Array" << endl;
for (int i = 0; i < size; i++)
{
    cout << Arr[i] << " ";
}
return 0;
}
```

**Output:**



## Searching:

### 1. Linear Searching:

```
#include <iostream>

using namespace std;

int Linear_Search(int Arr[], int size, int key)
{
    for (int i = 0; i < size; i++)
    {
        if (Arr[i] == key)
        {
            cout << "Match has found at index " << i << "." << endl;
        }
    }
    return -1;
}
```

```

int main()
{
    int size;

    cout << "Enter the size of Array" << endl;

    cin >> size;

    int Arr[size];

    cout << "Please Enter the values for filling the indexes of the array" << endl;

    for (int i = 0; i < size; i++)
    {
        cout << "Enter the value for index " << i << ": ";

        cin >> Arr[i];
    }

    cout << "Given Array" << endl;

    for (int i = 0; i < size; i++)
    {
        cout << Arr[i] << " ";
    }

    int key;

    cout << "\nPlease Enter the number you want to search in the array" << endl;

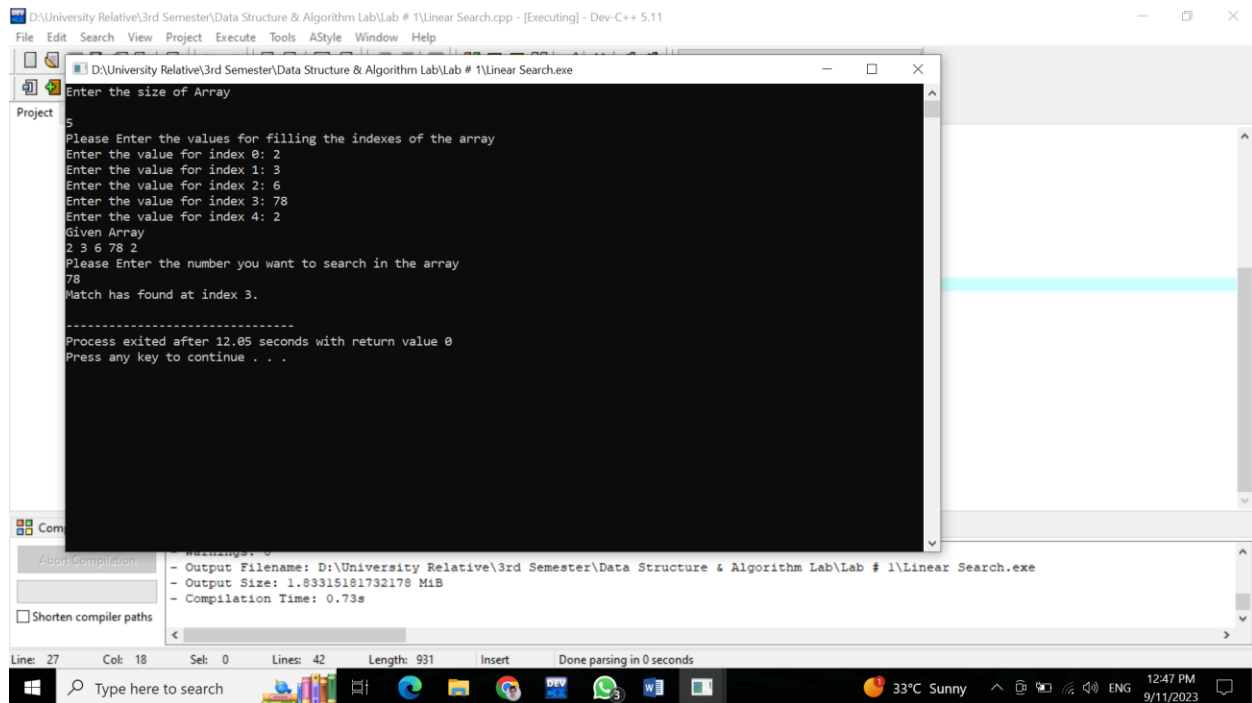
    cin >> key;

    int index = Linear_Search(Arr, size, key);

    return 0;
}

```

**Output:**



## 2. Binary Searching:

```
#include <iostream>
```

```
using namespace std;
```

```
int binarySearch(int A[], int n, int start, int end) {
```

```
    int loc = -1;
```

```
    while (start <= end) {
```

```
        int mid = (start + end) / 2;
```

```
        if (A[mid] == n) {
```

```
            loc = mid;
```

```
            break;
```

```
        } else if (n < A[mid]) {
```

```
            end = mid - 1;
```

```
        } else {
```

```
            start = mid + 1;
```

```
        }
```

```

    }
    return loc;
}

int main() {
    int size ;
    cout << "Enter the size of the array" << endl;
    cin >> size ;
    int Arr[size];
    for (int i = 0; i < size; i++)
    {
        cout << "Enter the value for index " << i << ": ";
        cin >> Arr[i];
    }
    int n, loc;
    loc = -1;
    cout << "Enter the number to find: ";
    cin >> n;
    loc = binarySearch(Arr, n, 0, 9);
    if (loc == -1) {
        cout << n << " not found!" << endl;
    } else {
        cout << n << " found at index " << loc << endl;
    }
    return 0;
}

```

**Output:**



D:\University Relative\3rd Semester\Data Structure & Algorithm Lab\Lab # 1\Binary Search.cpp - [Executing] - Dev-C++ 5.11

File Edit Search View Project Execute Tools AStyle Window Help

TDM-GCC 4.9.2 64-bit Release

D:\University Relative\3rd Semester\Data Structure & Algorithm Lab\Lab # 1\Binary Search.exe

```
Project Class
Enter the size of the array
5
Enter the value for index 0: 1
Enter the value for index 1: 9
Enter the value for index 2: 6
Enter the value for index 3: 5
Enter the value for index 4: 4
Enter the number to find: 4
4 found at index 4

-----
Process exited after 11.43 seconds with return value 0
Press any key to continue . . .
```

Compiler

About Compiler

Shorten compiler paths

- Output Size: 1.83315181732178 MiB  
- Compilation Time: 0.70s

search.exe

Line: 35 Col: 13 Sel: 0 Lines: 40 Length: 940 Insert Done parsing in 0.015 seconds

Type here to search

NIFTY +0.61%

12:49 PM 9/11/2023