**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++)

{

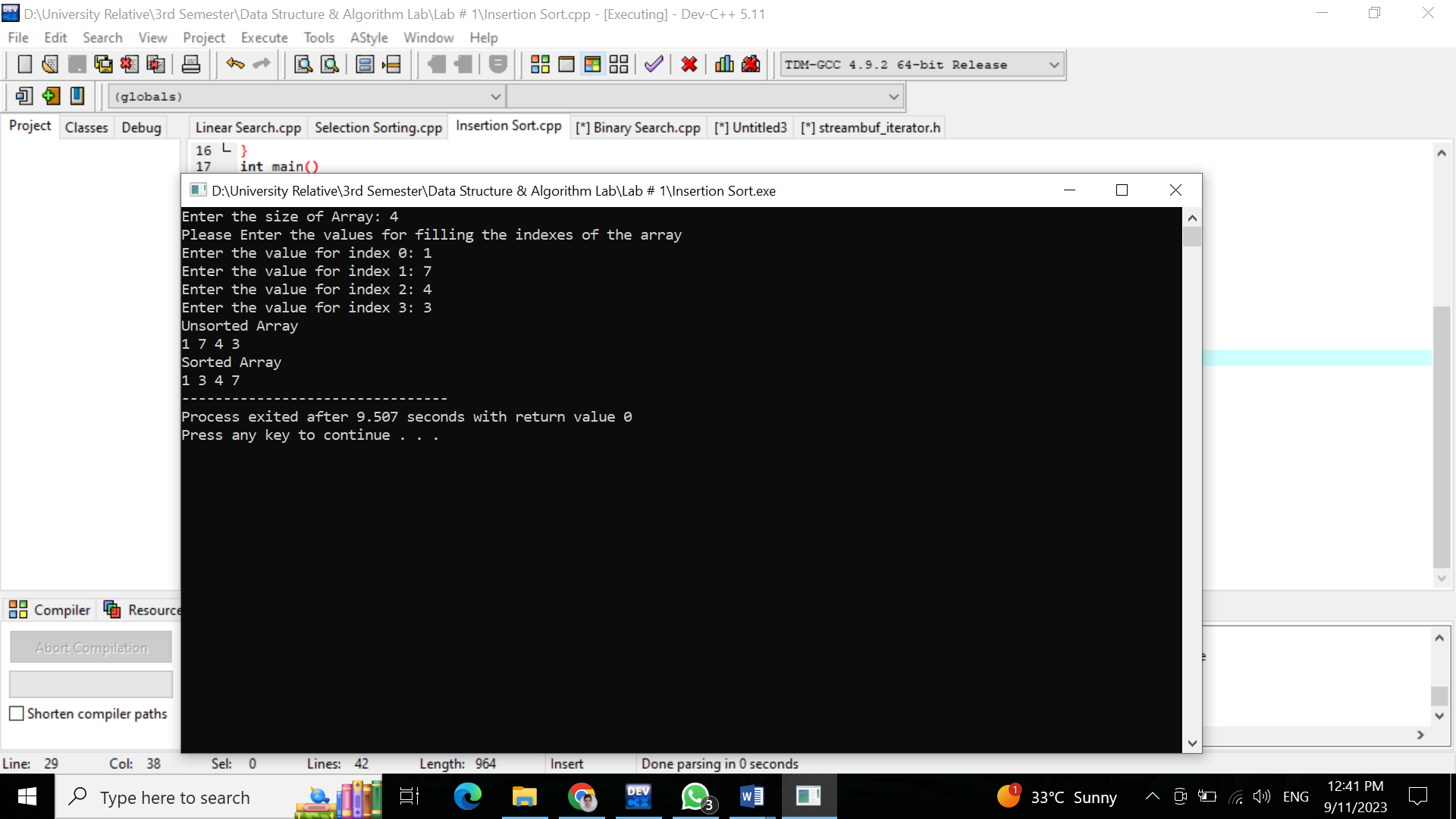
cout << Arr[i] << " ";

}

return 0;

}

**Output:**

****

**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++)

{

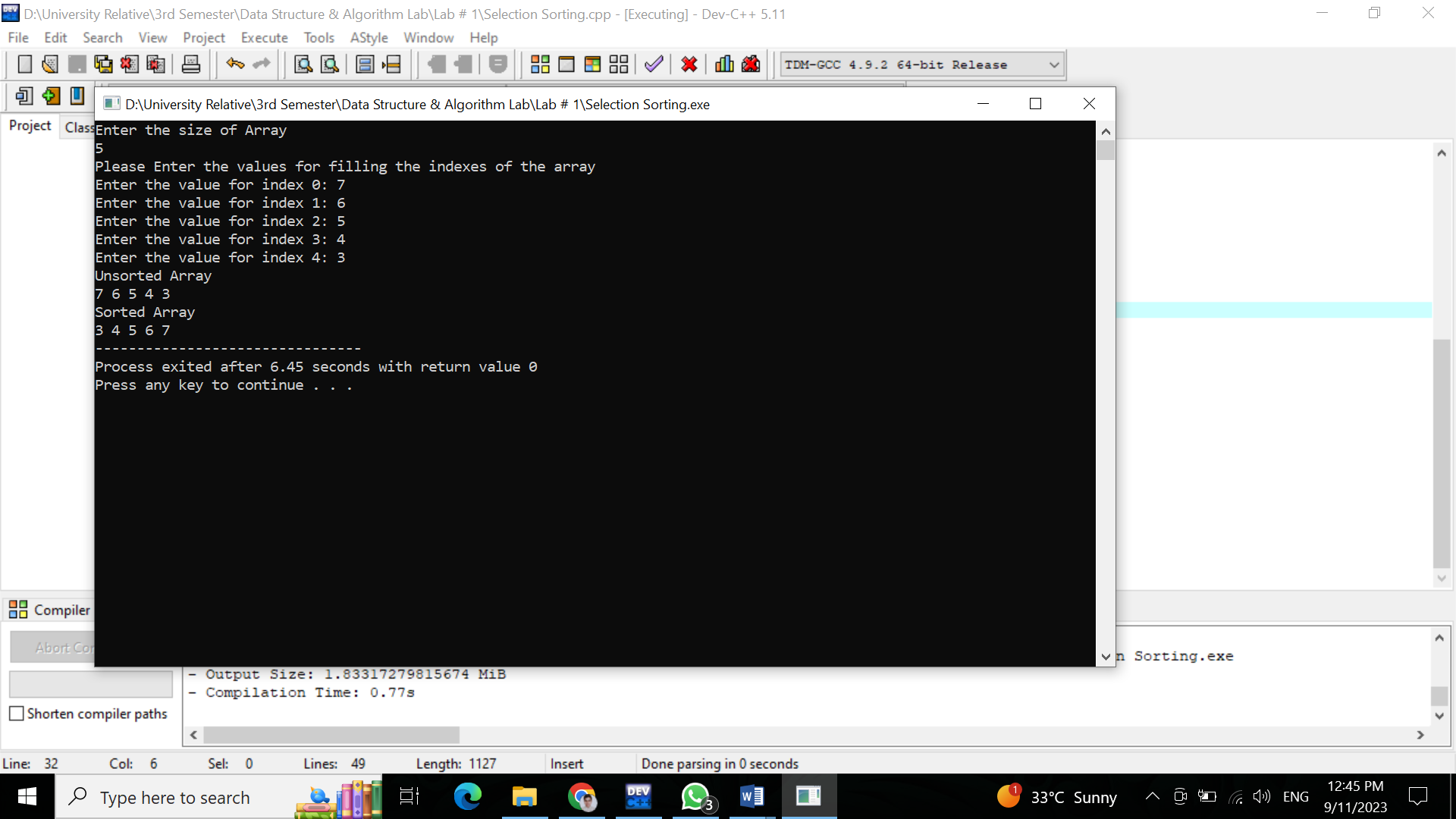
cout << Arr[i] << " ";

}

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;

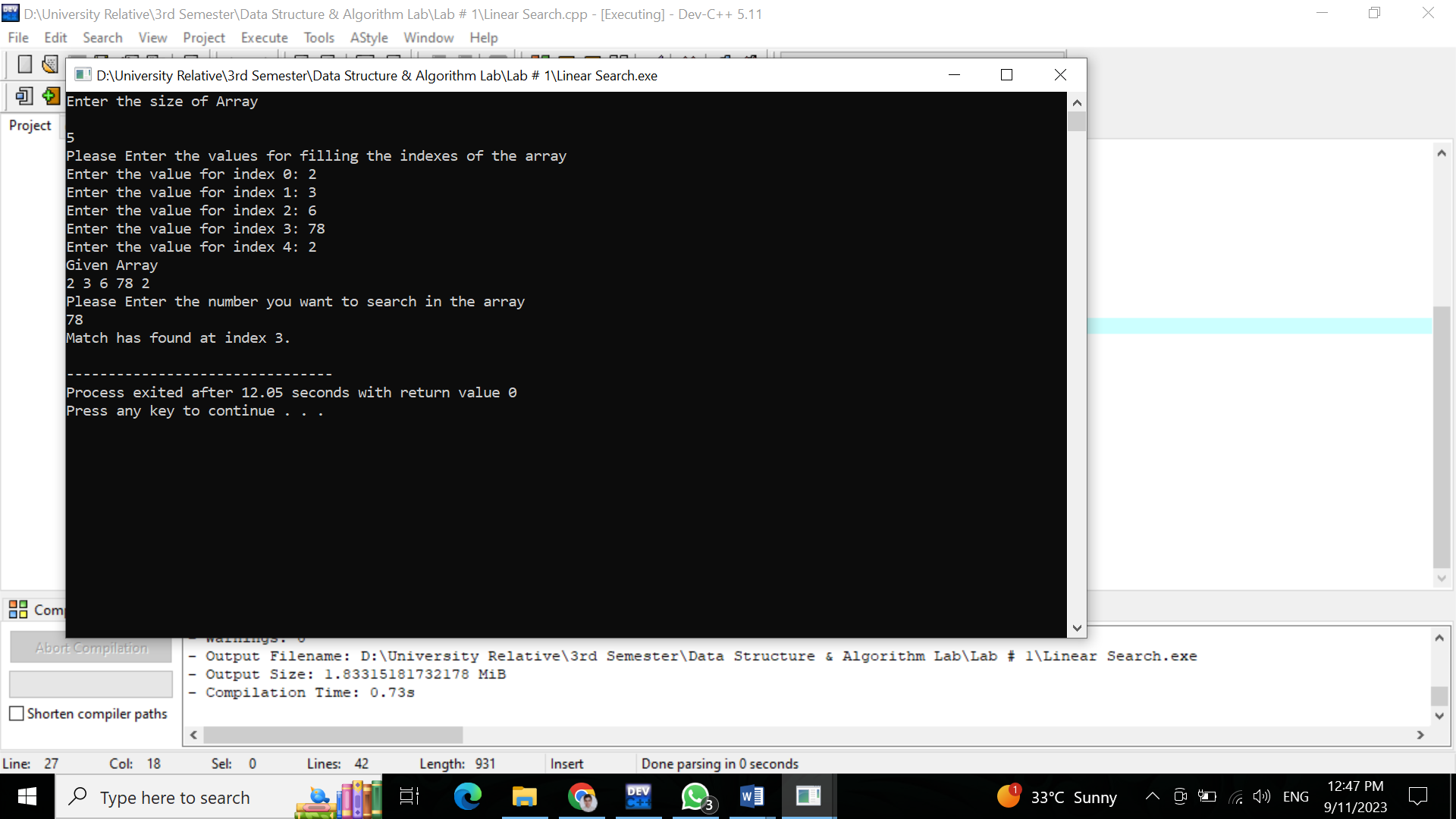
cin >> key;

int index = Linear\_Search(Arr, size, key);

return 0;

}

**Output:**

****

1. **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:**

