**Bubble Sort:**

Implementation of the code of Bubble Sort using User input Array.

**Code:**

#include<iostream>

using namespace std;

void bubbleSort(int arr[],int n){

for(int i=1;i<n;i++){

for (int j = 0; j < n-i; j++)

{

if (arr[j]>arr[j+1])

{

int temp = arr[j];

arr[j] = arr[j+1];

arr[j+1] = temp;

}

}

}

}

int main(){

int n;

cout << "\nEnter the Size of the Array: ";

cin >> n;

int arr[n];

cout << "\nEnter the elements from the Array: ";

for(int i=0;i<n;i++){

cin >> arr[i];

}

bubbleSort(arr,n);

for(int i=0;i<n;i++){

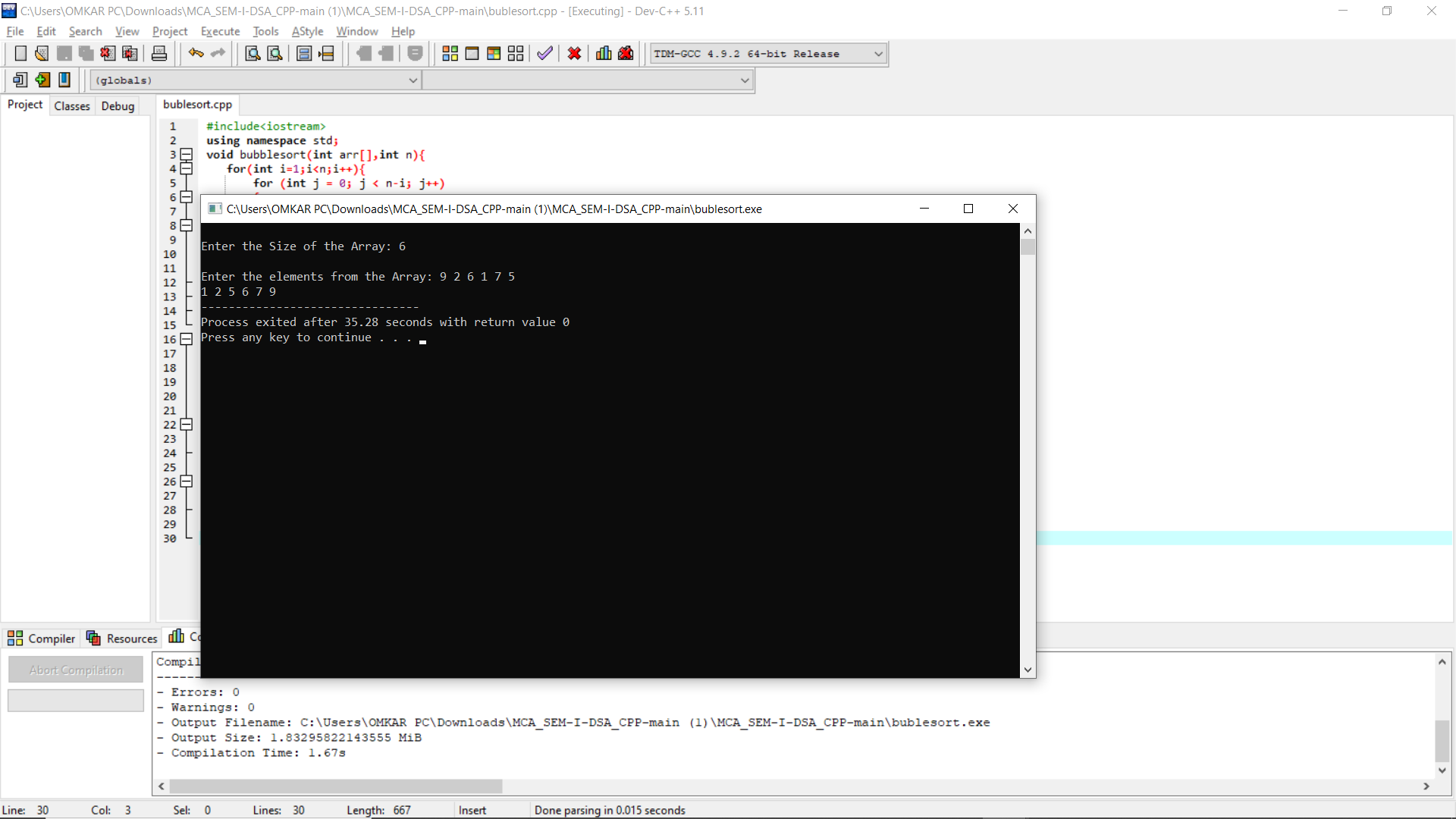
cout<< arr[i]<<" ";

}

return 0;

}

**Output:**



**Insertion Sort:**

Implementation of the code of Insertion Sort using User input Array.

**Code:**

#include <iostream>

using namespace std;

void insertionSort(int arr[], int n)

{

int i, key, j;

for (i = 1; i < n; i++)

{

key = arr[i];

j = i - 1;

while (j >= 0 && arr[j] > key)

{

arr[j + 1] = arr[j];

j = j - 1;

}

arr[j + 1] = key;

}

}

void printArray(int arr[], int n)

{

int i;

for (i = 0; i < n; i++)

cout << arr[i] << " ";

cout << endl;

}

int main()

{

int N;

cout << "\nEnter the size of Array: ";

cin >> N;

int arr[N];

cout << "\nEnter the elements of the Array: ";

for(int i = 0; i < N; i++){

cin >> arr[i];

}

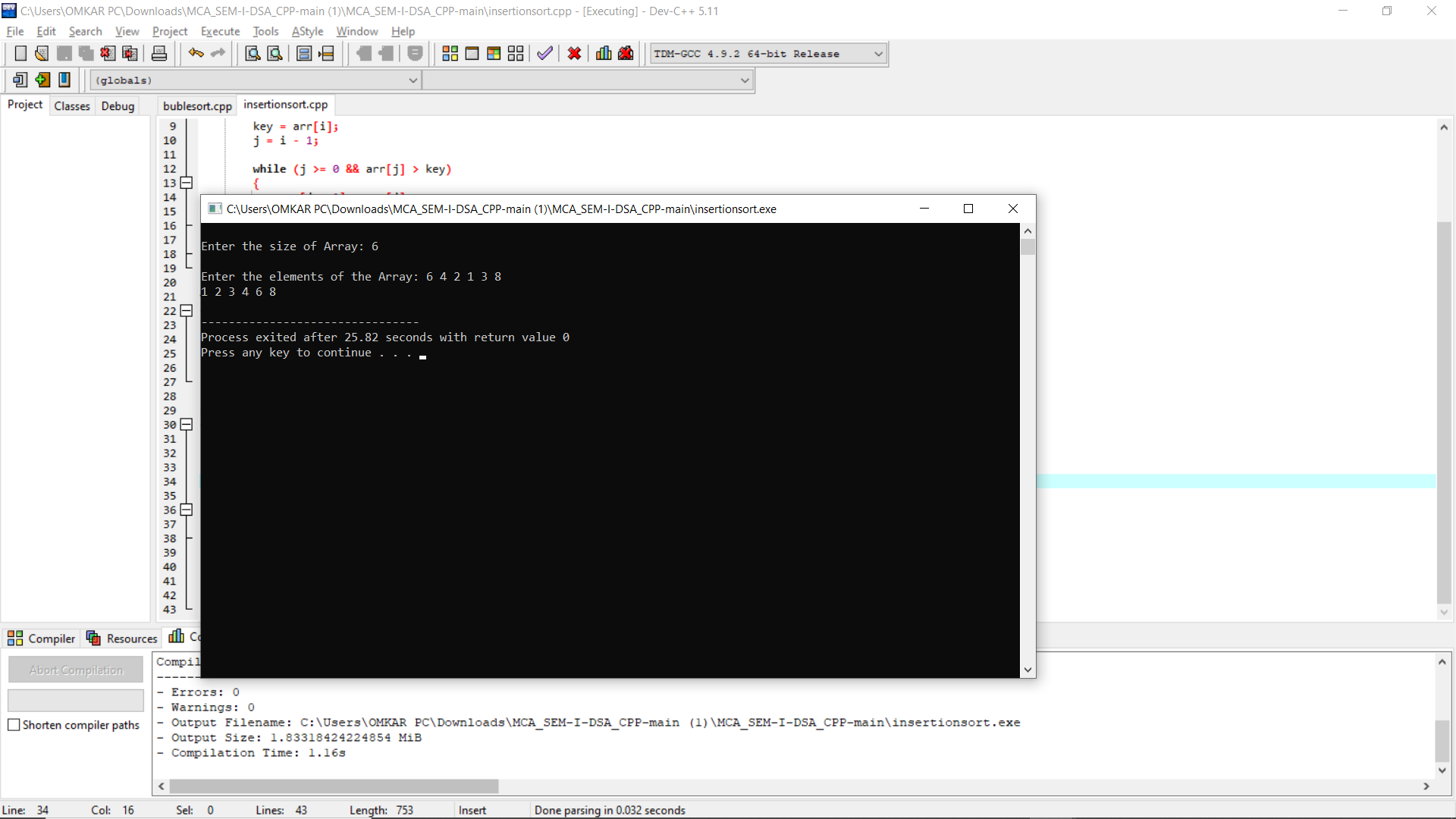
insertionSort(arr, N);

printArray(arr, N);

return 0;

}

**Output:**

****

**Selection Sort:**

Implementation of the code of Selection Sort using User input Array.

**Code:**

#include <iostream>

using namespace std;

void printArray(int array[], int size) {

for (int i = 0; i < size; i++) {

cout << array[i] << " ";

}

cout << endl;

}

void selectionSort(int array[], int size) {

for (int step = 0; step < size - 1; step++) {

int min\_idx = step;

for (int i = step + 1; i < size; i++) {

if (array[i] < array[min\_idx])

min\_idx = i;

}

int temp = array[min\_idx];

array[min\_idx] = array[step];

array[step] = temp;

}

}

int main() {

int size;

cout << "\nEnter the Size of the Array: ";

cin >> size;

int data[size];

cout << "\nEnter the Elements of the Array: ";

for (int i = 0; i < size; i++) {

cin >> data[i];

}

selectionSort(data, size);

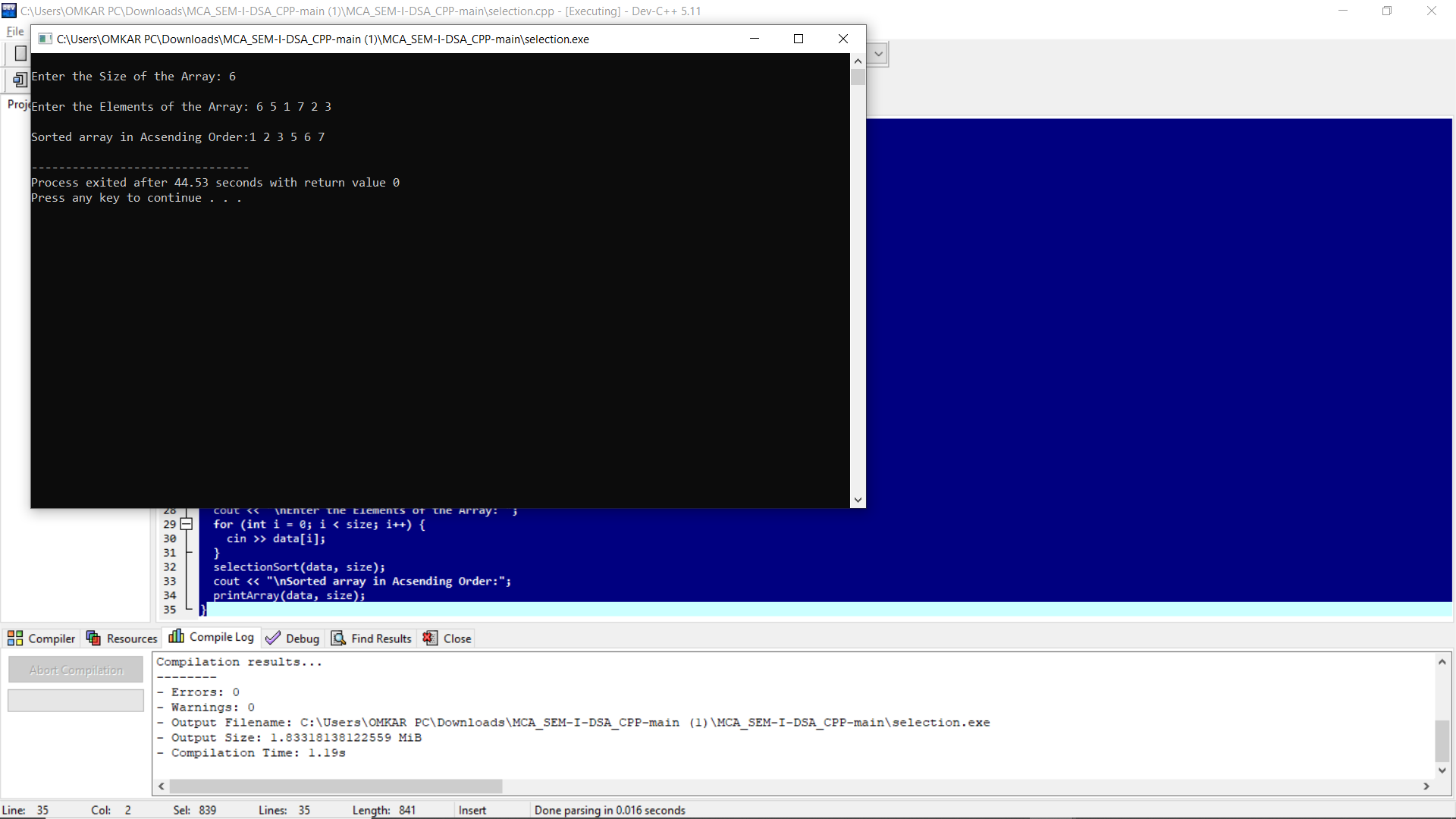
cout << "\nSorted array in Ascending Order:";

printArray(data, size);

return 0;

}

**Output:**

****

**Radix Sort:**

Implementation of the code of Radix Sort using User input Array.

**Code:**

#include <iostream>

using namespace std;

int getMax(int arr[], int n)

{

int mx = arr[0];

for (int i = 1; i < n; i++)

if (arr[i] > mx)

mx = arr[i];

return mx;

}

void countSort(int arr[], int n, int exp)

{

int output[n];

int i, count[10] = { 0 };

for (i = 0; i < n; i++)

count[(arr[i] / exp) % 10]++;

for (i = 1; i < 10; i++)

count[i] += count[i - 1];

for (i = n - 1; i >= 0; i--) {

output[count[(arr[i] / exp) % 10] - 1] = arr[i];

count[(arr[i] / exp) % 10]--;

}

for (i = 0; i < n; i++)

arr[i] = output[i];

}

void radixsort(int arr[], int n)

{

int m = getMax(arr, n);

for (int exp = 1; m / exp > 0; exp \*= 10)

countSort(arr, n, exp);

}

void print(int arr[], int n)

{

for (int i = 0; i < n; i++)

cout << arr[i] << " ";

}

int main()

{

int size;

cout << "\nEnter the Size of the Array: ";

cin >> size;

int data[size];

cout << "\nEnter the Elements of the Array: ";

for (int i = 0; i < size; i++) {

cin >> data[i];

}

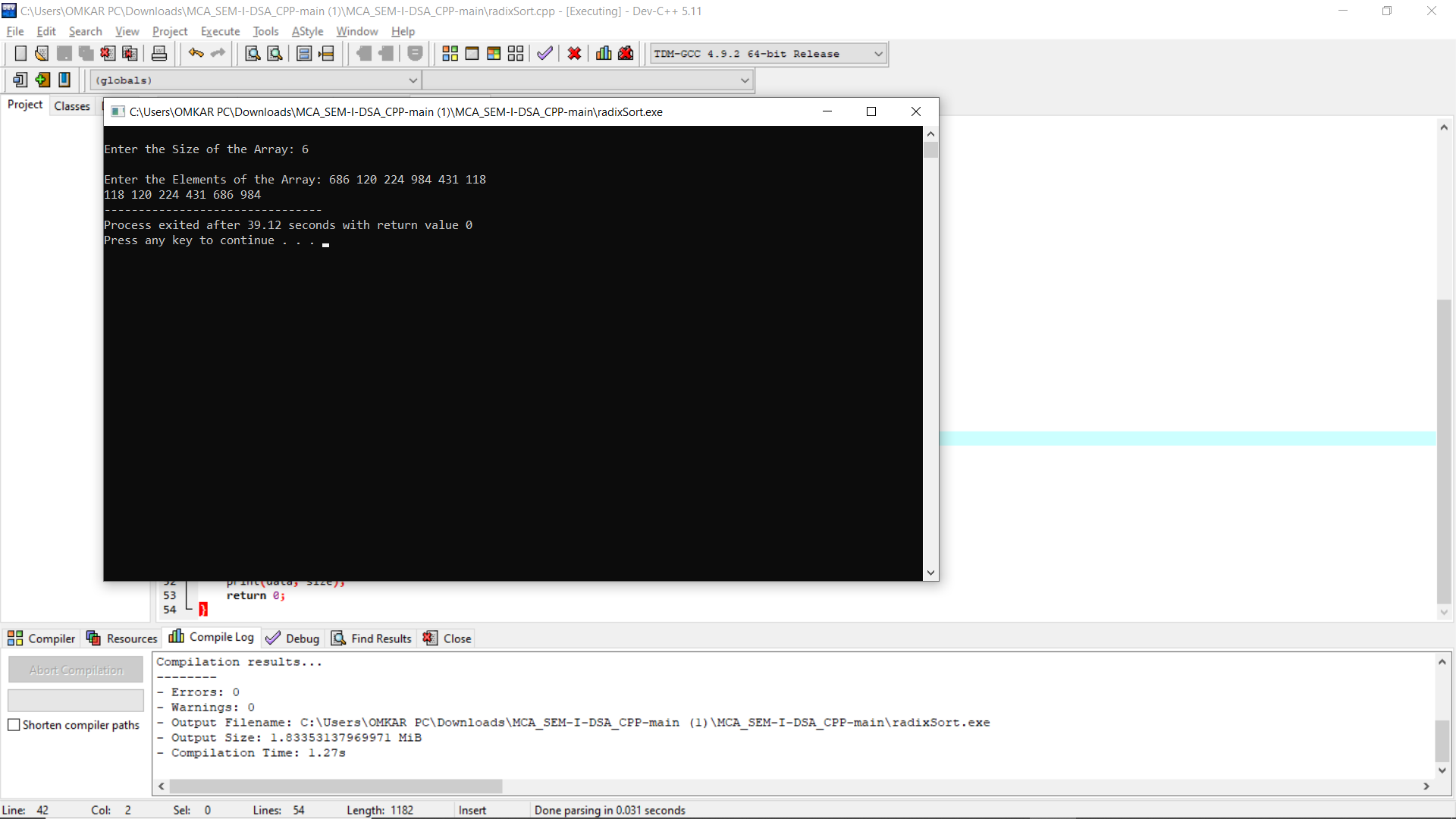
radixsort(data, size);

print(data, size);

return 0;

}

**Output:**

****

**Shell Sort:**

Implementation of the code of Shell Sort using User input Array.

**Code:**

#include <iostream>

using namespace std;

int shellSort(int arr[], int n)

{

for (int gap = n/2; gap > 0; gap /= 2)

{

for (int i = gap; i < n; i += 1)

{

int temp = arr[i];

int j;

for (j = i; j >= gap && arr[j - gap] > temp; j -= gap)

arr[j] = arr[j - gap];

arr[j] = temp;

}

}

return 0;

}

void printArray(int arr[], int n)

{

for (int i=0; i<n; i++)

cout << arr[i] << " ";

}

int main()

{

int n;

cout << "\nEnter the size of Array: ";

cin >> n;

int arr[n];

cout << "\nEnter the elements of the Array: ";

for(int i = 0; i < n; i++){

cin >> arr[i];

}

cout << "Array before sorting: \n";

printArray(arr, n);

shellSort(arr, n);

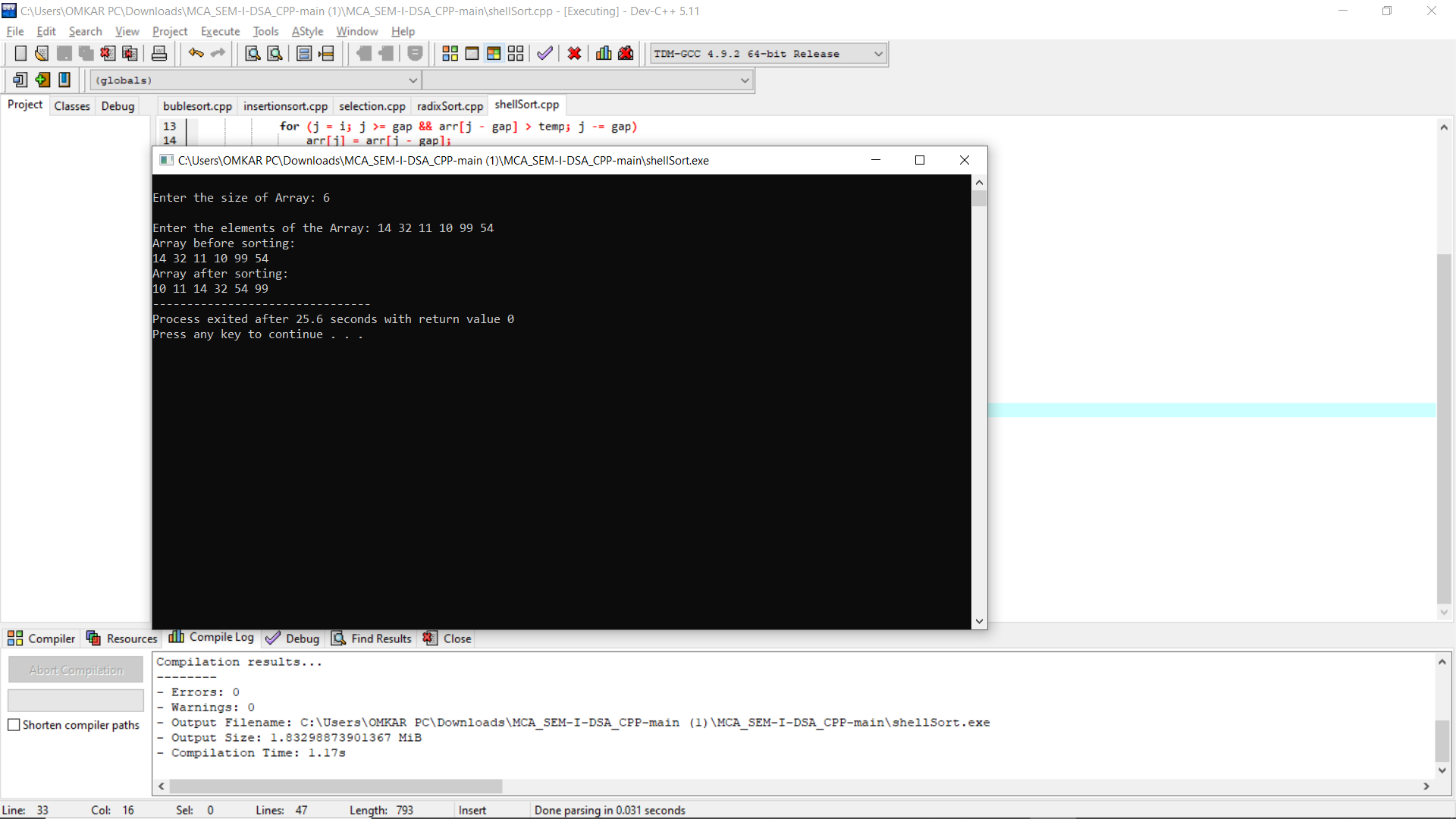
cout << "\nArray after sorting: \n";

printArray(arr, n);

return 0;

}

**Output:**

****