**QUESTION 1 :**

#include <iostream>

using namespace std;

class Arr {

public: void findLargestSmallest(int arr[], int size) {

if (size == 0) {

cout << "Array empty " << endl;

return;

}

int smallest = 10000;

int largest = 1;

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

if (arr[i] < smallest)

smallest = arr[i];

if (arr[i] > largest)

largest = arr[i];

}

cout << "Smallest element: " << smallest << endl;

cout << "Largest element: " << largest << endl;

}

};

int main() {

Arr arrr;

int arr[] = {

1110,

7,

9,

333,

323,

1

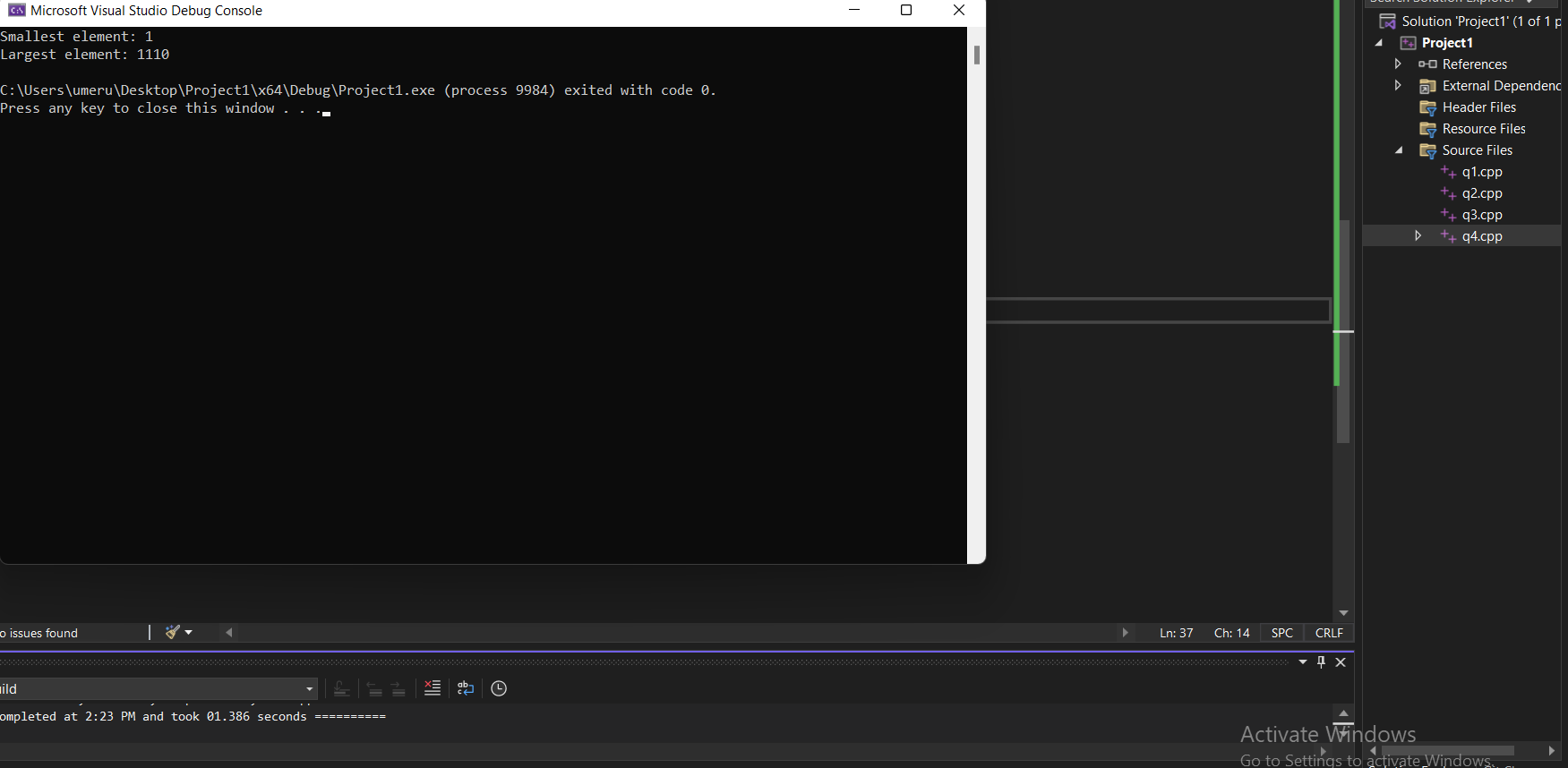
};

int size = sizeof(arr) / sizeof(arr[0]);

arrr.findLargestSmallest(arr, size);

return 0;

}

****

**QUESTION 2 :**

#include <iostream>

using namespace std;

class Arrayy {

public: void Productt(float arr[], int size) {

float product = 1.0;

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

arr[i] /= 2.0;

product \*= arr[i];

}

cout << "The product is: " << product << endl;

}

};

int main() {

Arrayy processor;

float arr[] = {

5.36,

31.7854,

1.43,

2.2,

62.21

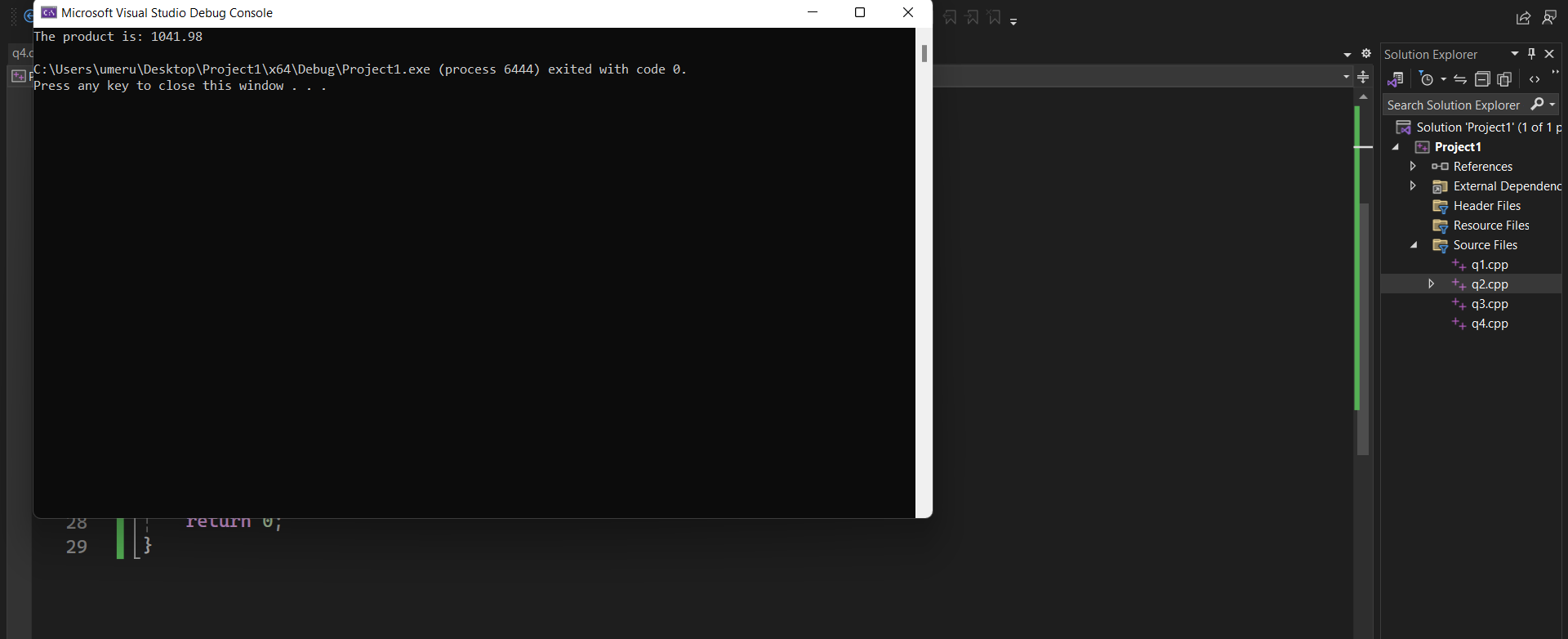
};

int size = sizeof(arr) / sizeof(arr[0]);

processor.Productt(arr, size);

return 0;

}



**QUESTION 3 :**

#include <iostream>

using namespace std;

class Arrayyy

{

public: void SimilarValues(int arr1[], int s1, int arr2[], int s2) {

cout << "Similar values ";

cout << "[ ";

for (int i = 0; i < s1; ++i) {

for (int j = 0; j < s2; ++j) {

if (arr1[i] == arr2[j]) {

cout << arr1[i] << " ";

break;

}

}

}

cout << " ]";

cout << endl;

}

};

int main() {

Arrayyy arr;

int array1[] = {

2,

4,

5,

6,

1

};

int size1 = sizeof(array1) / sizeof(array1[0]);

int array2[] = {

3,

6,

9,

0,

23,

55,

2

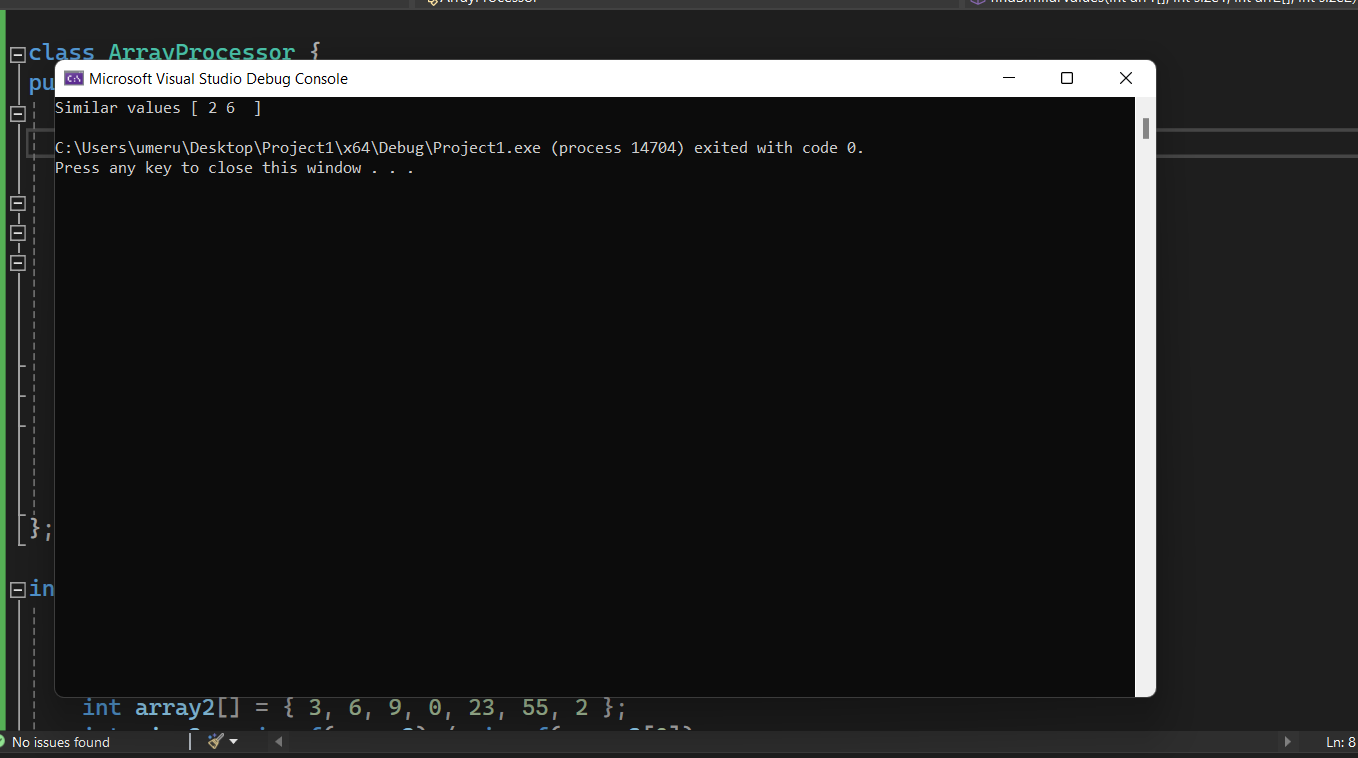
};

int size2 = sizeof(array2) / sizeof(array2[0]);

arr.SimilarValues(array1, size1, array2, size2);

return 0;

}



**QUESTION 4 :**

#include <iostream>

using namespace std;

class ArrayList {

private: int\* data;

int capacity;

int size;

public: ArrayList() {

capacity = 10;

size = 0;

data = new int[capacity];

}

~ArrayList() {

delete[] data;

}

void Adding(int value) {

if (size == capacity) {

int newCapacity = capacity \* 2;

int\* newData = new int[newCapacity];

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

newData[i] = data[i];

}

delete[] data;

data = newData;

capacity = newCapacity;

}

data[size++] = value;

}

void Removing(int index) {

if (index >= 0 && index < size) {

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

data[i] = data[i + 1];

}

size--;

}

}

};

int main() {

ArrayList List;

List.Adding(10);

List.Adding(20);

List.Adding(30);

List.Removing(1);

return 0;

}

