

Lab 6

BS-AI(22) | F22607017

LAB Task

Syed Zeeshan

2023

**Lab Tasks:**

**Task 1:**

**Q. Write a C++ program for Quick sort?**

**CODE:**

#include <iostream>

#include <vector>

using namespace std;

int partition(vector<int>& arr, int low, int high) {

int pivot = arr[high];

int i = low - 1;

for (int j = low; j < high; j++) {

if (arr[j] < pivot) {

i++;

swap(arr[i], arr[j]);

}

}

swap(arr[i + 1], arr[high]);

return i + 1;

}

void quickSort(vector<int>& arr, int low, int high) {

if (low < high) {

int pivotIndex = partition(arr, low, high);

quickSort(arr, low, pivotIndex - 1);

quickSort(arr, pivotIndex + 1, high);

}

}

void printArray(const vector<int>& arr) {

for (int num : arr) {

cout << num << " ";

}

cout << endl;

}

int main() {

vector<int> arr = { 12, 4, 5, 6, 7, 3, 1, 15 };

cout << "Original array: ";

printArray(arr);

int size = arr.size();

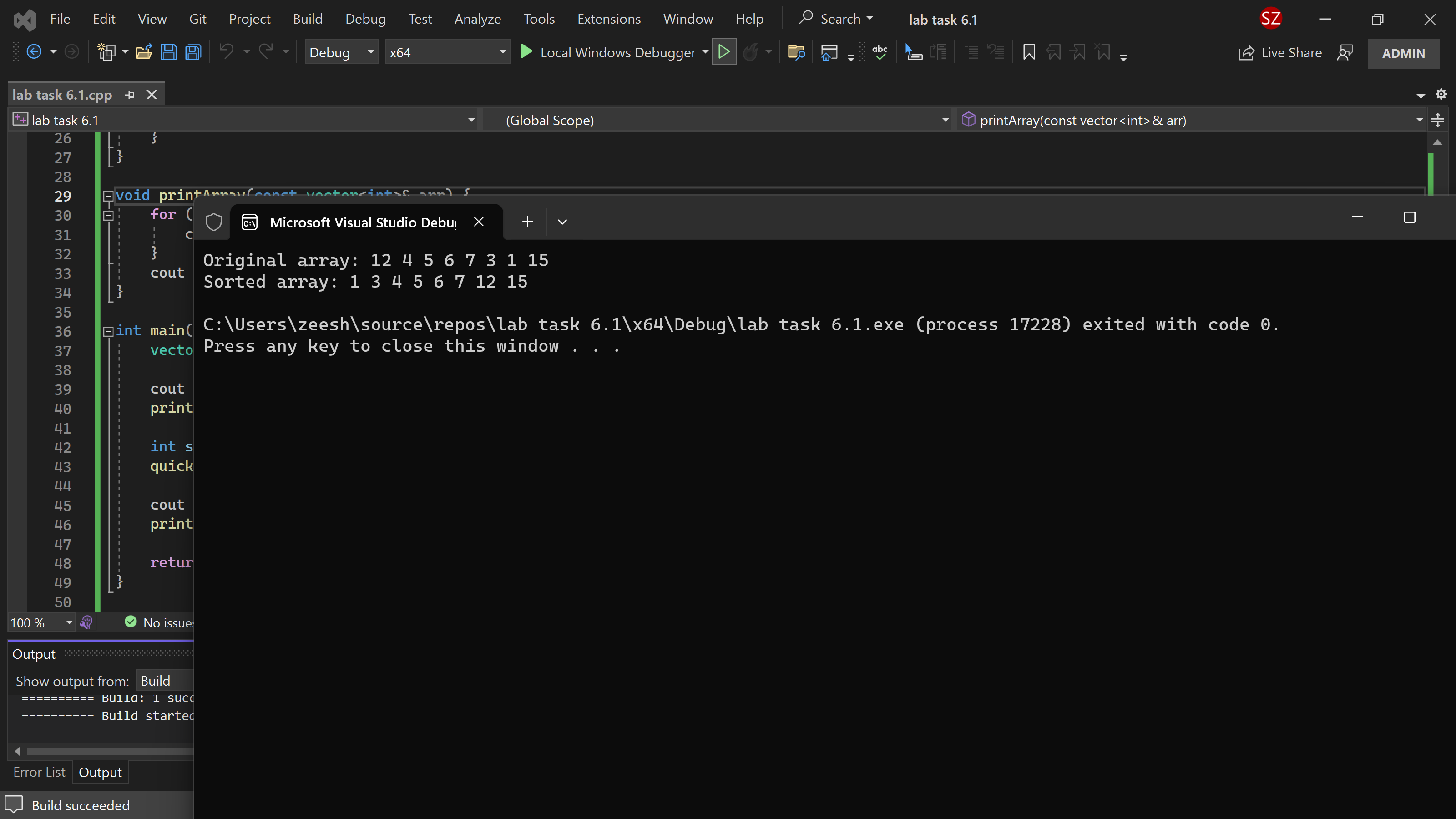
quickSort(arr, 0, size - 1);

cout << "Sorted array: ";

printArray(arr);

return 0;

}

OUTPUT:  


POSTLAB TASK:

Task 1:

Given a set of strings print the set sorted according to their length.

Input:

           The first line of input has an integer N that indicates the number of sets of strings, each set

may contain at least 1 and at most 10 strings, and each of the strings of the set may contain

between 1 and 20 inclusive characters (ONLY small English alphabet is allowed).

Output:

           The output should contain the set of input strings ordered by the length of strings. A blank

space must be printed between two words.

CODE:  
#include <iostream>

#include <vector>

#include <algorithm>

using namespace std;

int main() {

int N;

cin >> N;

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

vector<string> strings;

int numStrings;

cin >> numStrings;

// Read and store the strings in a vector

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

string str;

cin >> str;

strings.push\_back(str);

}

// Sort the vector of strings based on their length

sort(strings.begin(), strings.end(), [](const string& a, const string& b) {

return a.length() < b.length();

});

// Output the sorted set of strings

for (const string& str : strings) {

cout << str << " ";

}

cout << endl;

}

return 0;

}

OUTPUT:  
