EEE 102 C++ Programming and Software Engineering II

Assessment 1
SDP Report

Student: Kai-Yu Lu
ID: 1614649
Date: 2018.3.25

Problem statement:

This exercise requires students to write a function "bool same_vec (vector<int> a, vector<int> b)" to identify whether two vectors are identical (have the same elements, the order and multiplicities are all ignored). For example, we call {1, 5, 8, 13, 5, 8, 11, 1} and {1, 8, 8, 5, 1, 13, 13, 11} are identical.

The functions in this program should contain:

1. Users can use keyboard to input the elements for each

2. The length of each vector depends on the input that user have inputted.

3. Write two algorithms to solve this problem.

Model Answer – Software Development Process

Analysis

Inputs:

• Real numbers that users type, use space to add next element.

• Type "end" and press enter when finishing inputting.

Outputs:

• It will show whether two vectors are identical.

Design:

Algorithm 1

1. Create a Boolean method called "same_vec" which will be instanced in main whose name is called comparison. The descriptions of Boolean method will be supplied later.

2. a and b represent two vectors respectively.
   aElements and bElements represent the input which are type of string.

aE and bE can transform the types of aElements and bElements respectively by using stringstream.

3. Ask user to input the elements (only numbers), then type "end" and press enter when finishing inputting. The reason of typing "end" is to tell how to do the next step. In addition, when program gets "end", it will end corresponding function.

4. The basic idea of the first algorithm is to firstly sort two vectors. Then a method called "unique()" can pick up the same elements which are contiguous, and then move them to the end of the vector. Next, it is consequently delete the moved element. Then the rest elements in the corresponding vector are all unique and ordered. In conclusion, this process of this algorithm is:

   Input – Sort – Delete multiplicative elements which are contiguous - Output

5. Check the length of two vectors. If the lengths of two vectors were different, they would be evidently not identical. Next, use "for loop" to check whether the values in the same position in each vector are the same. If there was any difference, the result would be not identical, vice versa.

6. Display the sentence to tell users whether two vectors are identical.

Algorithm 2

1. Create a Boolean method called "same_vec" which will be instanced in main whose name is called comparison. The descriptions of Boolean method will be supplied later.

2. a and b represent two vector respectively.

aElements and bElements represent the input which are type of string. aE and bE can transform the types of aElements and bElements respectively by using stringstream.

3. Ask user to input the elements (only numbers), then type "end" and press enter when finishing inputting. The reason of typing "end" is to tell how to do the step next. In addition, when program gets "end", it will end corresponding function.

4. Use "for loop" to delete the same elements in each vector. Therefore, the rest elements in each vector are unique.

5. Check the length of two vectors. If the lengths of two vectors were different, they would be evidently not identical.

6. A and B are counters. When one of the elements in A find the same element in B, A will increase one time. When one of the elements in B find the same element in A, B will increase one time. If A is equal to the size of vecter a and B is equal to the size of vector b, two vectors are identical, vice versa. The reason of double check the sizes of A and B is to ensure that vector a could be s subset of vector b.

7. Display the sentence to tell users whether two vectors are identical.
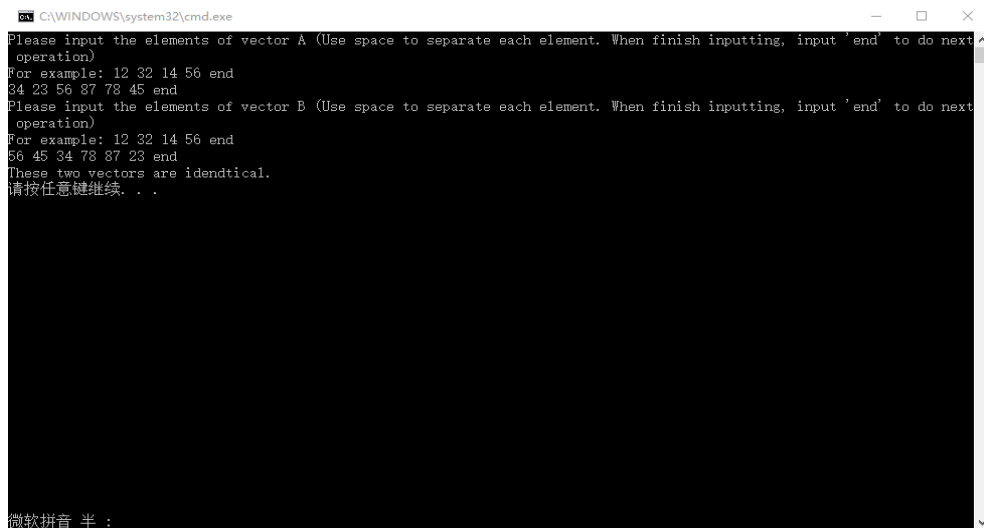
Implementation:

See the code for algorithm 1 in the file: exercise 1 for algorithm 1.cpp.
See the code for algorithm 2 in the file: exercise 1 for algorithm 2.cpp.
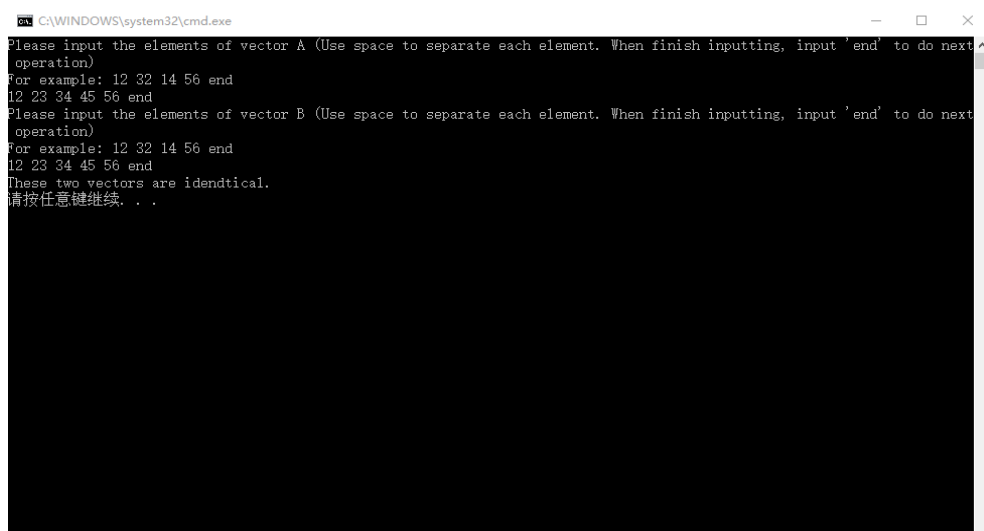
Testing:

For the result of Algorithm 1:

Figure 1: When two vectors have the same elements without ordered and multiplicative.

Figure 2: When two vectors have the same elements, which are ordered without multiplicative.

Figure 3: When two vectors the have same elements and multiplicative without ordered.

Figure 4: When two vectors the have different elements and multiplicative without ordered.

Figure 5: When two vectors the have different elements and ordered without
multiplicative.

Figure 6: When two vectors the have different elements without multiplicative
and ordered.

For the result of Algorithm 2:

The testing results are the same as the same as testing results of algorithm 1.