Student Name: Gao Jinliang
Student ID: 2022141450043

C++ Programming
Assignment 3

Section 1 Structured: Binding Using structured binding, the following function is implemented in just one line of in-function code:

```cpp
#include <iostream>
#include <string>
#include <map>
#include <functional>

using namespace std;

template <typename Key, typename Value, typename F>
void update(map<Key, Value>& m, F foo) {
    for (auto&& [key, value] : m) value = foo(value);
}

int main() {
    map<string, long long int> m {
            {"a", 1},
            {"b", 2},
            {"c", 3}
    };
    update(m, [](long long int value){
        return hash<string>{}(to_string(value));
    });
    for (auto&& [key, value] : m)
        cout << key << ":" << value << endl;
    return 0;
}
```

Section 2 References: Write a function that implements the exchange of two integers, requiring a reference implementation. Construct your own corresponding test program.

```cpp
#include <iostream>

using namespace std;

void exchange(int &a, int &b) {
    int temp = a;
    a = b;
    b = temp;
```

```
}

int main() {
    int a = 1;
    int b = 2;
    exchange(a, b);
    cout << "a = " << a << ", b = " << b << endl;
    return 0;
}
```

Section 3 Streams: Write a program to perform the following functions: Enter a series of student grades (name, grade) from the keyboard and write these student grades to the file stud.dat.

Code is as follows:

```
#include <iostream>
#include <fstream>
#include <string>

using namespace std;

class student{
private:
    string name;
    int score;
public:
    student(string& name, int& score) : name(name), score(score) {}
    void set_name(string& name){
        this->name = name;
    }
    void set_score(int& score){
        this->score = score;
    }
    string get_name(){
        return name;
    }
    int get_score(){
        return score;
    }
    void print() {
        cout << name << " " << score << endl;
    }
};

int main() {
    string name;
    int score;
    student stu0(name, score);
    ofstream fout("stu.dat", ios::binary);
    if (!fout) {
```

```
        cerr << "Failed to open file stud.dat" << endl;
        return -1;
    }

    int n;
    cout << "Enter the number of students: ";
    cin >> n;
    for (int i = 0; i < n; i++) {
        student s = student(name, score);
        cout << "Enter the name and score of student " << i + 1 << ": ";
        cin >> name >> score;
        s.set_name(name);
        s.set_score(score);
        fout.write(reinterpret_cast<char*>(&s), sizeof(student));
    }
    fout.close();

    ifstream fin("stu.dat", ios::binary);
    if (!fin) {
        cerr << "Failed to open file stu.dat" << endl;
        return -1;
    }
    cout << "The student scores are:" << endl;
    while (fin) {
        student s = student(name, score);
        fin.read(reinterpret_cast<char*>(&s), sizeof(student));
        if (fin.eof()) {
            break;
        }
        cout << s.get_name() << " " << s.get_score() << endl;
    }
    fin.close();

    return 0;
}
```

Running:

```
D:\Cpp\Streams\cmake-build-debug\Streams.exe
Enter the number of students: 3
Enter the name and score of student 1: James
99
Enter the name and score of student 2: Hebe 98
Enter the name and score of student 3: Tom 83
The student scores are:
James 99
Hebe 98
Tom 83
```

Section 4 STL(Containers):Keyboard input 5 integers, save these data into vector container, and use forward iterator and reverse iterator to iterate through the elements of vector and output them respectively.

Code is as follows:

```cpp
#include <iostream>
#include <vector>

using namespace std;

int main() {
    vector<int> v;
    int n;
    cout << "Enter 5 integers: ";
    for (int i = 0; i < 5; i++) {
        cin >> n;
        v.push_back(n);
    }

    cout << "Using forward iterator: ";
    for (auto it : v) {
        cout << it << " ";
    }
```

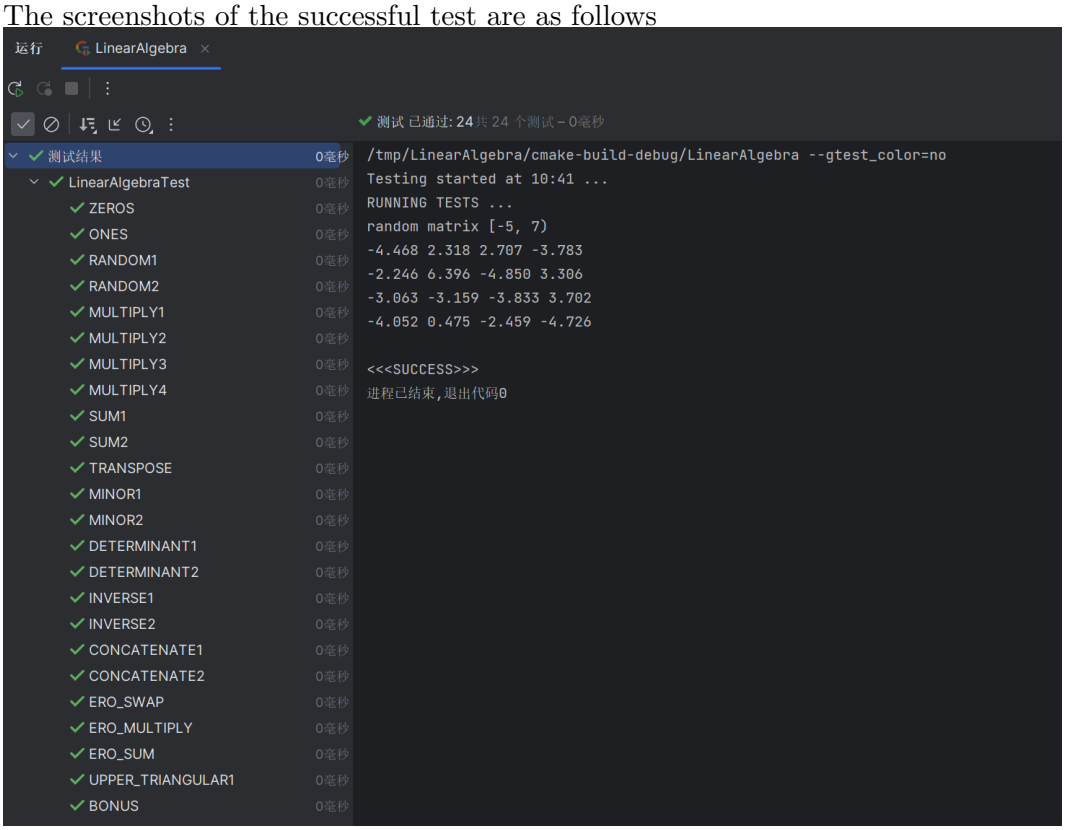
```
    cout << endl;

    cout << "Using reverse iterator: ";
    for (auto it = v.rbegin(); it < v.rend(); it++) {
        cout << *it << " ";
    }
    cout << endl;

    return 0;
}
```

Running:

```
D:\Cpp\VectorT\cmake-build-debug\VectorT.exe
Enter 5 integers: 1 2 3 4 5
Using forward iterator: 1 2 3 4 5
Using reverse iterator: 5 4 3 2 1
```

Section 5 Linear Algebra library

The screenshots of the successful test are as follows



```
root@fcf0e3c9bc0f:/# cd \ws
root@fcf0e3c9bc0f:/ws# cd LinearAlgebra
root@fcf0e3c9bc0f:/ws/LinearAlgebra# mkdir build
mkdir: cannot create directory 'build': File exists
root@fcf0e3c9bc0f:/ws/LinearAlgebra# cd build
root@fcf0e3c9bc0f:/ws/LinearAlgebra/build# cmake
Usage

  cmake [options] <path-to-source>
  cmake [options] <path-to-existing-build>
  cmake [options] -S <path-to-source> -B <path-to-build>

Specify a source directory to (re-)generate a build system for it in the
current working directory.  Specify an existing build directory to
re-generate its build system.

Run 'cmake --help' for more information.

root@fcf0e3c9bc0f:/ws/LinearAlgebra/build# cmake ..
-- Found Python: /usr/bin/python3.9 (found version "3.9.2") found components: Interpreter
-- Found Threads: TRUE
-- Configuring done
-- Generating done
-- Build files have been written to: /ws/LinearAlgebra/build
root@fcf0e3c9bc0f:/ws/LinearAlgebra/build# make
Scanning dependencies of target LinearAlgebra
[  8%] Building CXX object CMakeFiles/LinearAlgebra.dir/main.cpp.o
[ 16%] Building CXX object CMakeFiles/LinearAlgebra.dir/linearalgebra.cpp.o
[ 25%] Building CXX object CMakeFiles/LinearAlgebra.dir/src/unit_test.cpp.o
[ 33%] Linking CXX executable LinearAlgebra
[ 33%] Built target LinearAlgebra
Scanning dependencies of target gtest
[ 41%] Building CXX object _deps/googletest-build/googletest/CMakeFiles/gtest.dir/src/gtest-all.cc.o
[ 50%] Linking CXX static library ../../../lib/libgtest.a
[ 50%] Built target gtest
Scanning dependencies of target gmock
[ 58%] Building CXX object _deps/googletest-build/googlemock/CMakeFiles/gmock.dir/src/gmock-all.cc.o
[ 66%] Linking CXX static library ../../../lib/libgmock.a
[ 66%] Built target gmock
Scanning dependencies of target gmock_main
[ 75%] Building CXX object _deps/googletest-build/googlemock/CMakeFiles/gmock_main.dir/src/gmock_main.cc.o
[ 83%] Linking CXX static library ../../../lib/libgmock_main.a
[ 83%] Built target gmock_main
Scanning dependencies of target gtest_main
[ 91%] Building CXX object _deps/googletest-build/googletest/CMakeFiles/gtest_main.dir/src/gtest_main.cc.o
[100%] Linking CXX static library ../../../lib/libgtest_main.a
[100%] Built target gtest_main
```