

```
1  #include "STL_Containers.h"
2  #include <string>
3  #include <iostream>
4  #include <stdlib.h>
5  #include <ctime>
6  #include <cstdlib>
7  #include <algorithm>
8  #include <conio.h>
9
10
11 //=====
12 // Name      : STL_Containers.cpp
13 // Author    : Sotheanith Sok
14 // Version   : 1.0
15 // Description : This a tester used to test the STL Container by performing some function.
16 //=====
17
18 int main() {
19     std::srand((unsigned)time(0));
20     std::vector<int>v;
21     //Generate 10000 numbers between 1 and 100
22     STL_Containers::generateRandomNumbers(v);
23
24     //Print numbers in sorted order
25     STL_Containers::printSortedVector(v);
26
27     //Print the sum
28     std::cout << "\nThe sum of all number: " <<
        STL_Containers::calculateSumOfVector(v) << std::endl;
29
30     //Pause the program
31     std::cout << "\nPress any key to continue..." << std::endl;
32     _getch();
33
34     //Print the average
35     std::cout << "\nThe average of all number: " <<
        STL_Containers::calculateAverageOfVector(v) << "\n" << std::endl;
36
37     //Print the frequency
38     STL_Containers::printFrequency(v);
39     return 0;
40 }
41
42 //Precondition:
43 // _None.
44 //Postcondition:
45 // _Generate 10000 randoms number between 1 and 100 and put it into a vector.
46 void STL_Containers::generateRandomNumbers(std::vector<int>& v) {
```

```
47     for(int i =0;i<10000;++i)
48         v.push_back((std::rand() % 100)+1);
49 }
50
51 //Precondition:
52 // _None.
53 //Postcondition:
54 // _Print all value in a vector in sorted order.
55 void STL_Containers::printSortedVector(std::vector<int> v)
56 {
57     std::sort(v.begin(), v.end());
58     for (std::vector<int>::iterator it = v.begin(); it != v.end(); ++it) {
59         std::printf("%-4d", *it);
60     }
61     std::cout<<" "<<std::endl;
62 }
63
64
65 //Precondition:
66 // _None.
67 //Postcondition:
68 // _Calculate the sum of all values in a vector then return the sum.
69 int STL_Containers::calculateSumOfVector(std::vector<int>& v)
70 {
71     int total = 0;
72     for (std::vector<int>::iterator it = v.begin(); it != v.end(); ++it) {
73         total += *it;
74     }
75     return total;
76 }
77
78 //Precondition:
79 // _None.
80 //Postcondition:
81 // __Calculate the average of all values in a vector then return the average.
82 double STL_Containers::calculateAverageOfVector(std::vector<int>& v)
83 {
84     double avg = ((double)calculateSumOfVector(v))/v.size();
85     return avg;
86 }
87
88 //Precondition:
89 // _None.
90 //Postcondition:
91 // __Print the frequency of all values in a listed order.
92 void STL_Containers::printFrequency(std::vector<int>& v)
93 {
94     int array[100]{0};
95     for (std::vector<int>::iterator it = v.begin(); it != v.end(); ++it) {
```

```
96         array[( *it-1)] += 1;
97     }
98     printf( "%-6s%-3s%-8s\n", "Value", ":", "Frequency");
99     for (int i = 1; i <= 100; ++i) {
100         printf( "%-6d%-3s%-8d\n", i, ":", array[i-1] );
101     }
102 }
103
104
```