```cpp
#include "STL_Containers.h"
#include <string>
#include <iostream>
#include <stdlib.h>
#include <ctime>
#include <cstdlib>
#include <algorithm>
#include <conio.h>


//============================================================================
// Name        : STL_Containers.cpp
// Author      : Sotheanith Sok
// Version     : 1.0
// Description : This a tester used to test the STL Container by performing some function.
//============================================================================

int main() {
    std::srand((unsigned)time(0));
    std::vector<int>v;
    //Generate 10000 numbers between 1 and 100
    STL_Containers::generateRandomNumbers(v);

    //Print numbers in sorted order
    STL_Containers::printSortedVector(v);

    //Print the sum
    std::cout << "\nThe sum of all number: " << STL_Containers::calculateSumOfVector(v) << std::endl;

    //Pause the program
    std::cout << "\nPress any key to continue..." << std::endl;
    _getch();

    //Print the average
    std::cout << "\nThe average of all number: " << STL_Containers::calculateAverageOfVector(v) <<"\n"<< std::endl;

    //Print the frequency
    STL_Containers::printFrequency(v);
    return 0;
}

//Precondition:
//  _None.
//Postcondition:
//  _Generate 10000 randoms number between 1 and 100 and put it into a vector.
void STL_Containers::generateRandomNumbers(std::vector<int>& v) {
    for(int i =0;i<10000;++i)
        v.push_back((std::rand() % 100)+1);
}

//Precondition:
//  _None.
//Postcondition:
//  _Print all value in a vector in sorted order.
void STL_Containers::printSortedVector(std::vector<int> v)
{
    std::sort(v.begin(), v.end());
    for (std::vector<int>::iterator it = v.begin(); it != v.end(); ++it) {
        std::printf("%-4d", *it);
    }
    std::cout<<""<<std::endl;
}


//Precondition:
//  _None.
//Postcondition:
//  _Calculate the sum of all values in a vector then return the sum.
int STL_Containers::calculateSumOfVector(std::vector<int>& v)
{
    int total = 0;
    for (std::vector<int>::iterator it = v.begin(); it != v.end(); ++it) {
        total += *it;
    }
    return total;
}

//Precondition:
//  _None.
//Postcondition:
//  __Calculate the average of all values in a vector then return the average.
double STL_Containers::calculateAverageOfVector(std::vector<int>& v)
{
    double avg = ((double)calculateSumOfVector(v))/v.size();
    return avg;
}

//Precondition:
//  _None.
//Postcondition:
//  __Print the frequency of all values in a listed order.
void STL_Containers::printFrequency(std::vector<int>& v)
{
    int array[100]{0};
    for (std::vector<int>::iterator it = v.begin(); it != v.end(); ++it) {
        array[(*it-1)] += 1;
    }
    printf("%-6s%-3s%-8s\n", "Value",":", "Frequency");
    for (int i = 1; i <= 100;++i) {
        printf("%-6d%-3s%-8d\n",i,":",array[i-1] );
    }
}
```