CPP Coding Problem

Subject: Counter

Contributor: 林鼎傑,彭浩峰

Main testing concept: Operator, Template.

Basics	Functions
☐ C++ BASICS	☐ SEPARATE COMPILATION AND NAMESPACES
☐ FLOW OF CONTROL	☐ STREAMS AND FILE I/O
☐ FUNCTION BASICS	☐ RECURSION
☐ PARAMETERS AND OVERLOADING	■ INHERITANCE
☐ ARRAYS	■ POLYMORPHISM AND VIRTUAL FUNCTIONS
☐ STRUCTURES AND CLASSES	■ TEMPLATES
☐ CONSTRUCTORS AND OTHER TOOLS	☐ LINKED DATA STRUCTURES
■ OPERATOR OVERLOADING, FRIENDS, AND REFERENCES	☐ EXCEPTION HANDLING
☐ STRINGS	☐ STANDARD TEMPLATE LIBRARY
☐ POINTERS AND DYNAMIC ARRAYS	☐ PATTERNS AND UML

Description:

The counter counts how many items are in the counter.

- Please design "Counter" class and implement the following methods:

  - **Counter()**
    Constructs an empty counter.

  - **Counter(vector<T> items)**
    Constructs a counter by items.

  - **vector<T> most_common()**
    vector<T> most_common(int n)
    Return a list of the n most common elements ordered by the most common to the least. If n is negative or has no argument, then return all elements in the counter. Elements with equal counts are ordered in the order first encountered.

  - **int total()**
    Compute and return the sum of the counts, considering negative counts.

  - **void subtract()**
    void subtract(T key)
    void subtract(vector<T> keys)
    Element counts are subtracted 1 in certain keys or a key. If there are no arguments, all element counts will subtract 1. This operation will preserve results with counts of zero and less.

  - **Addition:**
    Implements the addition (define operator +) of two "Counter".
    Addition combines counters by adding the counts of corresponding elements. This operation can accept inputs with signed counts, but the output will exclude results with counts of zero or less, for example:
    There are 3 Counters c1, c2, c3,
    Initialize c1[1] = 2, c1[2] = 3 and c2[1] = 1, c2[2] = 4,

Let c3 = c1 + c2,
So c3[1] = 3, c3[2] = 7

- **Subtraction:**
  Implements the subtraction (define operator -) of two "Counter".

  Subtraction combines counters by subtracting the counts of corresponding elements. This operation can accept inputs with signed counts, but the output will exclude results with counts of zero or less, for example:
  There are 3 Counters c1, c2, c3,
  Initialize c1[1] = 2, c1[2] = 3 and c2[1] = 1, c2[2] = 4,
  Let c3 = c1 - c2,
  So c3[1] = 1

- **Promotion:**
  Implement a positive "Counter", i.e., "+Counter".
  (define operator +)
  Strips negative and zero counts

- **Negation:**
  Implement a negative "Counter", i.e., "-Counter".
  (define operator -)
  Strips positive and zero counts, and flips the sign on negative counts.

Note: **Counter must inherit map(*std::map<T, int>*).** Order of keys refer to std::map.
Negative counts are allowed.

Input:

No inputs.
**The main() function in your submission will be replaced when judging.
**You can use the main() function in "Sample Input" to test your program.

Output:

The result of executing your program with the given main function.

Sample Input / Output :

Sample Input	Sample Output

#include "Counter.h"
#include <iostream>
#include <vector>
 	
int main() {
  int myints[] = {1, 2, 3, 3, 2, 1, 2};
  std::vector<int> a(myints, myints + sizeof(myints) / sizeof(int));
  Counter<int> c(a);
  std::cout << c[99] << std::endl;
  for (auto i : c) {
  std::cout << i.first << ", " << i.second << std::endl;
  }
}
	0
	1, 2
	2, 3
	3, 2
	99, 0

☐ Eazy,Only basic programming syntax and structure are required.

■ Medium, Multiple programming grammars and structures are required. □ Hard, Need to use multiple program structures or more complex data types.
Expected solving time: 30 minutes
Other notes: Reference: Python documentation.