

Source File: ~/2336/28/lab28.cpp
Input: under control of main function
Output: under control of main function
Value: 2

Write a function template that will process the elements of a vector and determine the frequency of each element. Each distinct element and its frequency should be stored in an object of the class as specified in Figure 1. Maintain a **vector** of such objects. For each element in the vector, first determine if an entry for that element exists in the vector. Use the STL algorithm **find** to facilitate this determination. If an entry has previously been added to the vector, increment its frequency. If the element hasn't previously been added, add a new entry to the **back** of the vector.

A sample **main** function for testing your functions is shown in Figure 2. Commands to compile, link, and run this assignment are shown in Figure 3. To use the **Makefile** as distributed in class, add a target of **lab28main** to **targets1srcfile**.

```
1  #ifndef LAB28_H
2  #define LAB28_H
3
4  template<typename T>
5  class Frequency
6  {
7  public:
8      Frequency(T val) : value(val), frequency(1)
9      {}
10
11     void increment()
12     { ++frequency; }
13
14     T getValue() const
15     { return value; }
16
17     uint getFrequency() const
18     { return frequency; }
19
20     bool operator==(const T rhs) const;
21     bool operator< (const Frequency<T> rhs) const;
22 private:
23     T value;
24     uint frequency;
25 };
26
27 #endif
```

Figure 1. /usr/local/2336/include/lab28.h

```
1  #include <iostream>
2  #include <fstream>
3  #include <cstdlib>
4  #include <vector>
5  #include <algorithm>
6  #include <lab28.h>
7
8  using namespace std;
9
10 string ltrim(string s);
11 string rtrim(string s);
12
13 template<typename T>
14 vector<Frequency<T> > distribution(const vector<T>& v);
15
16 template<typename T>
17 ostream& operator<<(ostream& out, const vector<Frequency<T> >& v);
18
19 #include "lab28.cpp"
20
21 int main(int argc, char **argv)
22 {
23     const int aCount = 5, bCount = 7, cCount = 7, dCount = 12;
24     int a[aCount] = {5, 5, 5, 5, 5};
25     double b[bCount] = {7.7, 6.6, 5.5, 4.4, 3.3, 2.2, 1.1};
26     char c[cCount] = {'r', 'a', 'c', 'e', 'c', 'a', 'r'};
27     string d[dCount] = {"Cadillac", "GMC", "GMC",
28                         "Lexus", "Lexus", "Dodge", "GMC", "BMW",
29                         "BMW", "GMC", "Dodge", "Lexus"};
30     ifstream fin;
31
32     if (argc != 2)
33     {
34         cerr << "Usage: " << *argv << " InputFileName" << endl;
35         exit(EXIT_FAILURE);
36     }
37
38     fin.open(*(argv+1));
39     if (fin.fail())
40     {
41         cerr << "Couldn't open input file: " << *(argv+1) << endl;
42         exit(EXIT_FAILURE);
43     }
44
45     vector<int> aVector(a, a + sizeof(a) / sizeof(a[0]));
46     cout << distribution(aVector) << endl;
47
48     vector<double> bVector(b, b + sizeof(b) / sizeof(b[0]));
49     cout << distribution(bVector) << endl;
```

Figure 2. /usr/local/2336/src/lab28main.C (Part 1 of 3)

```
50
51     vector<char> cVector(c, c + sizeof(c) / sizeof(c[0]));
52     cout << distribution(cVector) << endl;
53
54     vector<string> dVector(d, d + sizeof(d) / sizeof(d[0]));
55     vector<Frequency<string> > dVectorDist;
56     dVectorDist = distribution(dVector);
57     cout << "Before Sorting by Frequency" << endl << dVectorDist << endl;
58     sort(dVectorDist.begin(), dVectorDist.end());
59     cout << "After Sorting by Frequency" << endl << dVectorDist << endl;
60
61     vector<char> charVec;
62     vector<Frequency<char> > charVecDist;
63     char ch;
64
65     while (fin.get(ch))          // read the data char-by-char
66         charVec.push_back(ch);
67
68     charVecDist = distribution(charVec);
69     cout << "Before Sorting by Frequency" << endl << charVecDist << endl;
70     sort(charVecDist.begin(), charVecDist.end());
71     cout << "After Sorting by Frequency" << endl << charVecDist << endl;
72
73     fin.clear();                // forget we hit end-of-file earlier
74     fin.seekg(0, ios::beg);     // move to the beginning of the file
75
76     vector<string> wordVec;
77     vector<Frequency<string> > wordVecDist;
78     string word;
79
80     while (fin >> word)         // read the data word-by-word
81     {
82         word = ltrim(word);
83         word = rtrim(word);
84         if (word.length() > 0)
85             wordVec.push_back(word);
86     }
87
88     wordVecDist = distribution(wordVec);
89     cout << "Before Sorting by Frequency" << endl << wordVecDist << endl;
90     sort(wordVecDist.begin(), wordVecDist.end());
91     cout << "After Sorting by Frequency" << endl << wordVecDist << endl;
92
93     return EXIT_SUCCESS;
94 }
95
```

Figure 2. /usr/local/2336/src/lab28main.C (Part 2 of 3)

```

96 // Function ltrim removes whitespace and punctuation from the beginning
97 // of string s
98 string ltrim(string s)
99 {
100     if (s.length() == 0)
101         return s;
102     else if (isspace(*(s.begin())) || ispunct(*(s.begin())))
103         return ltrim(s.substr(1));
104     else
105         return s;
106 }
107
108 // Function rtrim removes whitespace and punctuation from the end
109 // of string s
110 string rtrim(string s)
111 {
112     if (s.length() == 0)
113         return s;
114     else if (isspace(*(s.end() - 1)) || ispunct(*(s.end() - 1)))
115         return rtrim(s.substr(0, s.length() - 1));
116     else
117         return s;
118 }

```

Figure 2. /usr/local/2336/src/lab28main.C (Part 3 of 3)

```

1  newuser@csunix ~> cd 2336
2  newuser@csunix ~/2336> ./getlab.ksh 28
3  * Checking to see if a folder exists for Lab 28. . .No
4  * Creating a folder for Lab 28
5  * Checking to see if Lab 28 has sample input and output files. . .Yes
6  * Copying input and output files for Lab 28
7    from folder /usr/local/2336/data/28 to folder ./28
8  * Checking to see if /usr/local/2336/src/lab28main.C exists. . .Yes
9  * Copying file /usr/local/2336/src/lab28main.C to folder ./28
10 * Checking to see if /usr/local/2336/include/lab28.h exists. . .Yes
11 * Copying file /usr/local/2336/include/lab28.h to folder ./28
12 * Copying file /usr/local/2336/src/Makefile to folder ./28
13 * Adding a target of lab28main to targets1srcfile
14 * Touching file ./28/lab28.cpp
15 * Edit file ./28/lab28.cpp in Notepad++
16 newuser@csunix ~/2336> cd 28
17 newuser@csunix ~/2336/28> ls
18 01.dat      01.out      Makefile     lab28.cpp    lab28.h      lab28main.C
19 newuser@csunix ~/2336/28> make lab28main
20 g++ -g -Wall -std=c++11 -c lab28main.C -I/usr/local/2336/include -I.
21 g++ -o lab28main lab28main.o -L/usr/local/2336/lib -lm -lbits

```

Figure 3. Commands to Compile, Link, & Run Lab 28 (Part 1 of 3)

<pre> 22 newuser@csunix ~/2336/28> cat 01.dat 23 1992 1993 1994 24 This is a test of your word analysis program. 25 How many words did you find that begin with a vowel? 26 WHAT ABOUT STARTING WITH AN S OR ENDING IN AN s? 27 Mary said, "I like C++." </pre>	
<pre> 28 newuser@csunix ~/2336/28> ./lab28main 01.dat 29 5 -> 5 30 31 7.7 -> 1 32 6.6 -> 1 33 5.5 -> 1 34 4.4 -> 1 35 3.3 -> 1 36 2.2 -> 1 37 1.1 -> 1 38 39 r -> 2 40 a -> 2 41 c -> 2 42 e -> 1 43 44 Before Sorting by Frequency 45 Cadillac -> 1 46 GMC -> 4 47 Lexus -> 3 48 Dodge -> 2 49 BMW -> 2 50 51 After Sorting by Frequency 52 Cadillac -> 1 53 Dodge -> 2 54 BMW -> 2 55 Lexus -> 3 56 GMC -> 4 57 58 Before Sorting by Frequency 59 1 -> 3 60 9 -> 6 61 2 -> 1 62 -> 38 63 3 -> 1 64 4 -> 1 65 66 -> 5 67 T -> 6 68 h -> 3 69 i -> 9 </pre>	<pre> 70 s -> 8 71 a -> 9 72 t -> 5 73 e -> 4 74 o -> 8 75 f -> 2 76 y -> 5 77 u -> 2 78 r -> 6 79 w -> 5 80 d -> 6 81 n -> 4 82 l -> 3 83 p -> 1 84 g -> 2 85 m -> 2 86 . -> 2 87 H -> 3 88 b -> 1 89 v -> 1 90 ? -> 2 91 W -> 2 92 A -> 5 93 B -> 1 94 O -> 2 95 U -> 1 96 S -> 2 97 R -> 2 98 I -> 5 99 N -> 6 100 G -> 2 101 E -> 1 102 D -> 1 103 M -> 1 104 , -> 1 105 " -> 2 106 k -> 1 107 C -> 1 108 + -> 2 109 110 After Sorting by Frequency 111 B -> 1 </pre>

Figure 3. Commands to Compile, Link, & Run Lab 28 (Part 2 of 3)

112 U -> 1	155 9 -> 6	198 like -> 1
113 v -> 1	156 o -> 8	199 C -> 1
114 b -> 1	157 s -> 8	200
115 E -> 1	158 a -> 9	201 After Sorting by Frequency
116 D -> 1	159 i -> 9	202 that -> 1
117 M -> 1	160 -> 38	203 begin -> 1
118 , -> 1	161	204 with -> 1
119 4 -> 1	162 Before Sorting by Frequency	205 vowel -> 1
120 3 -> 1	163 1992 -> 1	206 WHAT -> 1
121 k -> 1	164 1993 -> 1	207 ABOUT -> 1
122 2 -> 1	165 1994 -> 1	208 STARTING -> 1
123 C -> 1	166 This -> 1	209 WITH -> 1
124 p -> 1	167 is -> 1	210 S -> 1
125 m -> 2	168 a -> 2	211 OR -> 1
126 . -> 2	169 test -> 1	212 ENDING -> 1
127 ? -> 2	170 of -> 1	213 IN -> 1
128 W -> 2	171 your -> 1	214 s -> 1
129 O -> 2	172 word -> 1	215 Mary -> 1
130 S -> 2	173 analysis -> 1	216 said -> 1
131 R -> 2	174 program -> 1	217 I -> 1
132 G -> 2	175 How -> 1	218 like -> 1
133 " -> 2	176 many -> 1	219 C -> 1
134 + -> 2	177 words -> 1	220 1992 -> 1
135 g -> 2	178 did -> 1	221 1993 -> 1
136 f -> 2	179 you -> 1	222 1994 -> 1
137 u -> 2	180 find -> 1	223 This -> 1
138 l -> 3	181 that -> 1	224 is -> 1
139 h -> 3	182 begin -> 1	225 test -> 1
140 l -> 3	183 with -> 1	226 of -> 1
141 H -> 3	184 vowel -> 1	227 your -> 1
142 n -> 4	185 WHAT -> 1	228 word -> 1
143 e -> 4	186 ABOUT -> 1	229 analysis -> 1
144	187 STARTING -> 1	230 program -> 1
145 -> 5	188 WITH -> 1	231 How -> 1
146 I -> 5	189 AN -> 2	232 many -> 1
147 t -> 5	190 S -> 1	233 words -> 1
148 y -> 5	191 OR -> 1	234 did -> 1
149 A -> 5	192 ENDING -> 1	235 you -> 1
150 w -> 5	193 IN -> 1	236 find -> 1
151 N -> 6	194 s -> 1	237 AN -> 2
152 d -> 6	195 Mary -> 1	238 a -> 2
153 T -> 6	196 said -> 1	239
154 r -> 6	197 I -> 1	
240 newuser@csunix ~/2336/28> ./lab28main 01.dat > my.out 241 newuser@csunix ~/2336/28> diff 01.out my.out 242 newuser@csunix ~/2336/28>		

Figure 3. Commands to Compile, Link, & Run Lab 28 (Part 3 of 3)