```
Script started on 2023-07-27 12:23:54-05:00 [TERM="xterm" TTY="/dev/pts/2" COLUMNS=
bi94684@ares:~$ pwd
/home/students/bj94684
bi94684@ares:~$ cat box.info
    NAME: Jose Barron
                                                 CLASS: CSC122-002
           Lab: A box upon ye
                                                  Level: 5
         Option:
                                                  Level: + 0
                                            Total Level: 5.0
* This program is designed to create a frame around a phrase given by a
* user, either from keyboard input or a file. The user has the ability
* to choose how the words are centered (center, left, or right) and what *
* symbol the frame is made of. The words are processed with a vector of *
 * strings which are the words then with the use of the FramedWord class *
 * the words are framed.
bi94684@ares:~$ show-code strextra.h
strextra.h:
     1 #ifndef STREXTRA H INC
        #define STREXTRA H INC
     4 #include<iostream>
      #include<string>
       #include<vector>
        #include<cctype>
    8
    9
       using namespace std;
   10
    11
      inline std::string tolower str(std::string s)
    12
    13 {
    14
            std::string t = s:
    15
            for(std::string::size type pos = 0; pos != s.length(); ++pos)
    16
    17
                t[pos] = static cast<char>( tolower(t[pos]) );
    18
    19
            return t;
    20
    21 }
    22
       inline std::vector<std::string::size type> find(std::string s, char t,
    24
                                              bool case sensitive = true)
    25
    26
           if( ! case sensitive)
    27
    28
                s = tolower str(s);
```

```
29
                t = static cast<char>( tolower(t) );
    30
            }
    31
            std::vector<string::size type> num;
    32
            for (std::string::size type pos = 0; pos != s.length(); ++pos)
    33
    34
    35
                if (static cast<char>( s[pos] ) == t )
    36
    37
                    num.push back(pos);
    38
    39
    40
            return num:
    41
    42
        inline bool str is incl(std::string s, std::string t, std::string::size ty;
    44
                                                     std::string::size type end)
    45
    46
            std::string nr = s.substr(beg, end-beg+1);
    47
            bool tf;
    48
            if ( nr == t)
    49
                return tf = true;
    50
    51
    52
            return tf = false:
    53 }
    54
        inline std::string::size type strcom(std::vector<string::size type> b,
                                         std::vector<string::size type> e,
    56
    57
                                         std::string s, std::string t)
    58
    59
            string::size type num;
    60
            for(auto p : b)
    61
    62
                for(auto p1 : e)
    63
    64
                                    // The starter position can never be greater
                    if (p < p1)
    65
    66
                        bool ys = str is incl(s, t, p, p1);
    67
                        if (ys)
    68
    69
                            return num = p:
    70
    71
                    }
    72
    73
    74
            return num = s.length();
    75
    76
    77
        std::string::size type find(std::string s, std::string t, bool c s = true)
    78
    79
    80
    81 #endif
bj94684@ares:~$ show-code strextra.h
```

```
strextra.h:
     1 #ifndef STREXTRA H INC
        #define STREXTRA H INC
     4 #include<iostream>
     5
       #include<string>
       #include<vector>
        #include<cctvpe>
     7
     8
    9
        using namespace std;
    10
    11
    12 inline std::string tolower str(std::string s)
    13 {
    14
            std::string t = s;
    15
            for(std::string::size type pos = 0; pos != s.length(); ++pos)
    16
    17
                t[pos] = static cast<char>( tolower(t[pos]) );
    18
    19
            return t:
    20
    21 }
    22
        inline std::vector<std::string::size type> find(std::string s, char t,
    23
                                              bool case sensitive = true)
    24
    25
    26
            if( ! case sensitive)
    27
    28
                s = tolower str(s);
    29
                t = static cast<char>( tolower(t) );
    30
            }
    31
    32
            std::vector<string::size type> num;
    33
            for (std::string::size type pos = 0; pos != s.length(); ++pos)
    34
    35
                if (static cast<char>( s[pos] ) == t )
    36
    37
                    num.push back(pos);
    38
    39
    40
            return num;
    41 }
    42
    43 inline bool str is incl(std::string s, std::string t, std::string::size ty;
    44
                                                     std::string::size type end)
    45 {
    46
            std::string nr = s.substr(beg, end-beg+1);
    47
            bool tf;
    48
            if ( nr == t)
    49
            {
```

```
50
                return tf = true;
    51
            }
    52
            return tf = false;
    53 }
    54
    55 inline std::string::size type strcom(std::vector<string::size type> b,
    56
                                        std::vector<string::size type> e,
    57
                                        std::string s, std::string t)
    58 {
    59
            string::size type num;
    60
            for(auto p : b)
    61
    62
                for(auto p1 : e)
    63
                                    // The starter position can never be greater
    64
                    if (p < p1)
    65
    66
                        bool ys = str is incl(s, t, p, p1);
    67
                        if (ys)
    68
    69
                            return num = p;
    70
    71
                    }
    72
    73
    74
            return num = s.length();
    75 }
    76
    77
        std::string::size type find(std::string s, std::string t, bool c s = true)
    78
    79
    80
    81 #endif
bj94684@ares:~$ show-code box.cpp
box.cpp:
     1 #include <iostream>
     2 #include <fstream>
       #include <vector>
     4 #include <cctvpe>
       #include <limits>
       #include "strextra.h"
        using namespace std;
    9
    10
       inline string::size type find space( string s, char t)
    11
    12
            string::size type num = s.length();
    13
            for (string::size type pos = 0; pos < s.length(); ++pos)</pre>
    14
    15
                if (static cast<char>( s[pos] ) == t )
    16
```

```
17
                return num = pos;
18
            }
19
        }
20
        return num:
21 }
22
23
   vector<string> word get(string sen);
24
25
   class FrameWord
   {
26
27 private:
28
        vector<string> words:
29
        short max length word;
30
        vector<string> framed words;
31
        vector<string> framed words left:
        vector<string> framed words right;
32
33
        char frame c;
34
35
        void center word(string & word) const;
36
        void center word left(string & word) const;
37
        void center word right(string & word) const;
38
39
40 public:
41
        FrameWord(const vector<string> & inputted words, char f c);
        void frame words();
42
43
        void frame words left();
44
        void frame words right();
45
        void display frame() const;
46
        void display frame left() const;
47
        void display frame right() const;
48
49 };
50
51
   FrameWord::FrameWord(const vector<string> & inputted words, char f c)
52
                        : words(inputted words), max length word(0), frame c(f
53
54
55
        for (const auto & word : words)
56
57
            max length word = max(max length word, static cast<short>(word.length)
58
59 }
60
   void FrameWord::center word(string & word) const
61
62 {
63
        short space = max length word - static cast<short>(word.length());
64
        short left side = space / 2:
        short right side = space - left side;
65
        word = frame c + string(left side + 1, ' ') + word + string(right side
66
67 }
68
69 void FrameWord::center word left(string & word) const
70 {
```

```
71
         short space = max length word - static cast<short>(word.length());
 72
         word = frame c + word + string(space + 2, ' ') + frame c:
 73 }
 74
 75
    void FrameWord::center word right(string & word) const
 76
 77
         short space = max length word - static cast<short>(word.length());
 78
         word = frame c + string(space + 1, ' ') + word + " " + frame c;
 79
 80
     void FrameWord::frame words()
 82
 83
         for (const auto & word : words)
 84
 85
             string framed word = word;
 86
             center word(framed word);
 87
             framed words.push back(framed word);
 88
         }
 89
    }
 90
     void FrameWord::frame words left()
 93
         for (const auto & word : words)
 94
 95
             string framed word left = word;
 96
             center word left(framed word left);
 97
             framed words left.push back(framed word left);
 98
 99
     }
100
101 void FrameWord::frame words right()
102
103
         for (const auto & word : words)
104
105
             string framed word right = word;
106
             center word right(framed word right);
107
             framed words right.push back(framed word right);
108
         }
109
110
111
     void FrameWord::display frame() const
112
113
         short frame size = max length word + 4;
114
         cout << string(frame size, frame c) << '\n';</pre>
115
         for (const auto & word : framed words)
116
         {
117
             cout << word << endl;</pre>
118
119
120
         cout << string(frame size, frame c) << '\n';</pre>
121 }
122
123 void FrameWord::display frame left() const
124 {
```

```
125
         short frame size = max length word + 4;
126
         cout << string(frame size, frame c) << '\n':</pre>
127
         for (const auto & word : framed words left)
128
129
             cout << word << endl:</pre>
130
131
         cout << string(frame size, frame c) << '\n';</pre>
132 }
133
134 void FrameWord::display frame right() const
135 {
136
         short frame size = max length word + 4:
137
         cout << string(frame size, frame c) << '\n';</pre>
138
         for (const auto & word : framed words right)
139
140
             cout << word << endl;</pre>
141
142
         cout << string(frame size, frame c) << '\n';</pre>
143 }
144
145 int main()
146 {
147
         cout << "\n\t\tA Box Upon Ye! Program\n\n";</pre>
148
         bool done = false:
149
         vector<string> words;
150
151
         {
152
             char choice;
             cout << "Are you inputting from keyboard or file?: ";</pre>
153
154
155
             choice = static cast<char>( toupper( choice ) );
             cin.ignore(numeric limits<streamsize>::max(), '\n');
156
157
             if ( choice == 'F')
158
159
                  ifstream file;
160
                  string fn;
161
                  cout << "\nPlease enter the name of your names files: ";</pre>
162
                  getline(cin, fn);
163
                  file.open(fn);
164
                  while ( ! file )
165
166
                      file.close();
                      file.clear();
167
168
                      cout << "\nIm sorry I could open " << '"' << fn << '"'</pre>
169
                          << ". Please enter another name: ";
170
                      getline(cin, fn);
171
                      file.open(fn);
172
                  cout << "\nFile " << '"' << fn << '"' << " was opened succesful
173
174
                  string s:
175
                 while (!file.eof() )
176
177
                      getline(file, s);
178
                      words.push back(s);
```

```
179
180
                  file.close():
181
                  file.clear();
182
                  done = true:
183
184
             else if ( choice == 'K')
185
186
                  cout << "\nInsert Sentence: ";</pre>
187
                  string sen;
188
                  getline(cin,sen);
189
                 words = word get(sen);
190
                  done = true:
191
             else
192
193
194
                  cout << "You did not input any correct answer."</pre>
195
                       << "Please Try Again\n";
196
197
198
199
         } while ( ! done ):
200
201
         bool done1 = false:
202
         bool center = false:
203
         bool left = false;
204
         bool right = false;
205
206
         do
         {
207
208
209
             cout << "\nHow do you want to center the word? (Center, Left, or R:</pre>
210
             cin >> choice;
211
             choice = static cast<char>( toupper( choice ) );
212
             cin.ignore(numeric limits<streamsize>::max(), '\n');
213
             if ( choice == 'C')
214
             {
215
                  center = true:
216
                  done1 = true;
217
218
             else if ( choice == 'L')
219
220
                  left = true:
221
                  done1 = true;
222
223
             else if ( choice == 'R')
224
225
                  right = true;
226
                  done1 = true:
227
228
             else
229
230
                  cout << "You did not input Center, Left, or Right."</pre>
                       << " Please Try Again";
231
232
             }
```

```
233
         } while ( ! done1 );
         char frame c;
234
235
         bool done2 = false;
236
237
238
             char t;
239
             cout << "\nWhat would you like your frame be made of? ";</pre>
240
241
             if ( isprint(t))
242
                 frame c = t:
243
244
                 done2 = true:
245
246
             else
247
248
                 cout << "You did not input a printable character."</pre>
249
                      << "Please Try Again";
250
251
         } while ( ! done2 );
252
253
         FrameWord FrameWord(words.frame c):
254
255
         if ( center)
256
257
             FrameWord.frame words();
258
             cout << "\nFramed Words:\n";</pre>
                      FrameWord.display frame();
259
260
             cout << '\n';
261
262
         else if (left)
263
264
             FrameWord.frame words left();
265
             cout << "\nFramed Words:\n":</pre>
                     FrameWord.display frame left();
266
267
             cout << '\n';
268
269
         else if (right)
270
271
             FrameWord.frame words right();
272
             cout << "\nFramed Words:\n":</pre>
273
                      FrameWord.display frame right();
274
             cout << '\n':
275
         }
276
         return 0;
277 }
278
279 vector<string> word get(string sen)
280 {
281
         vector<string> words;
282
         vector<string::size type> num = find(sen, ' ');
283
         string sen1 = sen:
284
         for( vector<string::size type>::size type p = 0; p != num.size(); ++p)
285
286
                 string::size type spaces = find space(sen1, ' ');
```

```
287
                    if ( spaces != sen1.length())
   288
   289
                        string word = sen1.substr(0,spaces);
   290
                        words.push back(word):
   291
                        sen1 = sen\overline{1}.substr(spaces + 1. sen1.length()):
   292
   293
   294
            string word1 = sen.substr(num[num.size() - 1] + 1, sen.length());
            words.push back(word1);
   295
   296
            return words:
   297 }
bi94684@ares:~$ CPP strextra box
box.cpp***
strextra.cpp...
box.cpp: In constructor
'FrameWord::FrameWord(const
std::vector<std:: cxx11::basic string<char> >&, char)':
box.cpp:51:1: warning:
'FrameWord::framed words' should be initialized
in the member initialization list [-Weffc++]
   51 | FrameWord::FrameWord(const vector<string>
   & inputted words, char f c)
      ^~~~~~~
box.cpp:51:1: warning:
'FrameWord::framed words left'
should be initialized in the member initialization list
[-Weffc++]
box.cpp:51:1: warning:
'FrameWord::framed words right'
should be initialized in the member initialization list
[-Weffc++]
bi94684@ares:~$ ./box.out
                A Box Upon Ye! Program
Are you inputting from keyboard or file?: sa
You did not input any correct answer. Please Try Again
Are you inputting from keyboard or file?: key
Insert Sentence: 0 | -+- | /-\
How do you want to center the word? (Center, Left, or Right)? s
You did not input Center, Left, or Right. Please Try Again
How do you want to center the word? (Center, Left, or Right)? center
What would vou like your frame be made of? X
Framed Words:
XXXXXXX
X 0 X
X \mid X
X -+- X
```

```
X | X
X /-\ X
XXXXXXX
bj94684@ares:~$ ./box.out
               A Box Upon Ye! Program
Are you inputting from keyboard or file?: keyboard
Insert Sentence: I went to eat food
How do you want to center the word? (Center, Left, or Right)? rightt
What would you like your frame be made of? +
Framed Words:
+++++++
+ I +
+ went +
   to +
+ eat +
+ food +
++++++
bj94684@ares:~$ ./box.out
               A Box Upon Ye! Program
Are you inputting from keyboard or file?: fi
Please enter the name of your names files: phrase
Im sorry I could open "phrase". Please enter another name: phrase.k
File "phrase.k" was opened succesfully
How do you want to center the word? (Center, Left, or Right)? left
What would you like your frame be made of? ^
Framed Words:
^^^^^
^hello ^
^world ^
^^^^
bj94684@ares:~$ ./box.out
               A Box Upon Ye! Program
Are you inputting from keyboard or file?: fileee
```

```
File "phrase1.k" was opened succesfully
How do you want to center the word? (Center, Left, or Right)? h
You did not input Center, Left, or Right. Please Try Again
How do you want to center the word? (Center, Left, or Right)? right
What would you like your frame be made of? *
Framed Words:
******
      the *
      sky *
      is *
* falling *
******
bj94684@ares:~$ ./box.out
               A Box Upon Ye! Program
Are you inputting from keyboard or file?: Key
Insert Sentence: the sky is falling
How do you want to center the word? (Center, Left, or Right)? Center
What would you like your frame be made of? @
Framed Words:
<u>ඉවෙනුවෙනුවෙනුව</u>
   the
   sky
  is
@ falling @
bj94684@ares:~$ exit
exit
Script done on 2023-07-27 12:27:34-05:00 [COMMAND EXIT CODE="0"]
```

Please enter the name of your names files: phrase1.k