```
1 /*********************
  * AUTHOR : JOSHUA SALZEDO, Chris Burrell
3 * LAB #8
            : Array Train - Set # 1
4 * CLASS
            : CS1A
  * SECTION
            : MW: 8AM
  * DUE DATE : 04/23/2018
  *****************************
8 #include <iostream>
9 #include <iomanip>
10 #include "MyHeader.h"
11 #include "functions.h"
12
13 /**************************
14
  * Array Train - Set # 1
  * -----
1.5
16 * This program outputs min, max, avg, and sum of two parallel arrays
17 * and outputs selected elements from user input
18 * -----
  * INPUT:
19
  * Age to saerch by
20
  * name so earch by
21
22
23 * OUTPUT:
24 * Average age , minimum age, maximum age, sum of ages,
25 * namees and ages from user input
27
28 int main()
29 {
30 /*******************
31 * CONSTANTS
32 * -----
33 * OUTPUT - USED FOR CLASS HEADING
  * ______
34
           : Programmer's Name
35
  * PROGRAMMER
35 ^ FNOC._
36 * CLASS
              : students course
              : Class Days and Times
38 * LAB NUM
              : Lab Number
39 * LAB NAME
               : Title of the lab
40 * ----
41
  * PROCESSING - USED FOR PROGRAM EXECUTION
42
43 * NAMES : hardcoded input names (parallel array)
44 * AGES : hardcoded input ages (parallel array)
45 * PROMPT FIND INT : Prompt for an age input
46 * PROMPT NAME : Prompt for a name input
48
    const char PROGRAMMER[] = "Joshua Salzedo, Chris Burrell";
    const char CLASS[] = "CS1A";
49
    const char SECTION[] = "MW: 8:00a - 12:00p";
50
51
    const int LAB NUM = 25;
52
    const char LAB NAME[] = "Array Train # 1";
53
54
    const int AR LEN = 11;
55
    const string NAMES[] = {
56
57
          "zac",
58
          "Kasra",
          "Bas",
59
          "Sara",
60
61
          "Nick",
```

```
62
                "Delvin",
 63
                "Justin",
 64
                "Abe",
 65
                "Jeremy",
                "Farah",
 66
                "Maryan",
 67
 68
        };
 69
        const int AGES[] = {
 70
               22,
 71
                75,
 72
                19,
 73
                21,
 74
                18,
 75
                12.
 76
                19,
 77
                5,
 78
                62,
 79
                21,
 80
                21
 81
       };
        const string PROMPT FIND INT = "Please enter an age you want to look "
 82
 83
                                       "for: ";
 84
        const string PROMPT NAME = "enter a name to find: ";
 85
 86
 87
       double averageAges;
                                       // CALC & OUT - average value of AGES
 88
                                            array
 89
       int firstInstanceOfAgeIndex;
                                       // CALC & OUT - index for First position
 90
                                           for input int
       //
 91
                                       // CALC & OUT - total occurences of
      int totalOccurrences;
 92
                                           an input int
       //
 93
       int nameIndex;
                                       // CALC & OUT - index for found name
 94
       int sumAges;
                                       // CALC & OUT - sum of AGES array
                                       // CALC & OUT -index of the smallest age
 95
       int youngestPersonIndex;
 96
      int oldestPersonIndex;
                                      // CALC & OUT - index of the biggest age
                                      // IN & OUT
 97
      int searchInt;
                                                      - user provided input
 98
      string searchString;
                                      // IN & OUT
                                                      - user provided input
99
      string formattedString;
                                      // CALC & OUT - Formatted output strings
                                     // CALC & OUT - name of oldest person
100
      string oldestPerson;
                                       // CALC & OUT - name of youngest person
101
       string youngestPerson;
102
103
       // OUTPUT - class header
       DisplayHeader (PROGRAMMER, CLASS, SECTION, LAB NUM, LAB NAME);
104
105
106
       // compute the average of the AGES array
107
        averageAges = AverageIntArray(AGES, AR LEN);
108
109
       // set up formatting
110
       cout << fixed << setprecision(2);</pre>
        cout << "Average value of ages array is : " << averageAges;</pre>
111
112
       // clean up after ourselves
113
       cout << scientific << setprecision(6) << endl << endl;</pre>
114
115
       // find an int to search for
        searchInt = SaneInputCinInt(0, 100, PROMPT FIND INT);
116
117
118
      // find the first instance of the searchInt
119
      firstInstanceOfAgeIndex = FindFirstInstance(AGES, AR LEN, searchInt);
120
121
       // check if an instance was actually found
122
        if (firstInstanceOfAgeIndex != AR LEN)
```

```
123
      {
           formattedString = "you found ";
124
125
           formattedString += NAMES[firstInstanceOfAgeIndex];
           formattedString += " who is ";
126
            formattedString += to string(AGES[firstInstanceOfAgeIndex]);
127
            formattedString += " years old at index # ";
128
129
            formattedString += to string(firstInstanceOfAgeIndex) += '\n';
130
131
132
       else
133
       {
134
            formattedString = "Name not found on the list.\n";
135
       cout << formattedString << endl;</pre>
136
137
138
      // find total occurences of the age
139
      totalOccurrences = FindOccurrencesInt(AGES, AR LEN, searchInt);
140
       // reinit
141
       formattedString = "";
142
143
       // switch over # occurences, for word choice and verb tense.
144
      switch (totalOccurrences)
145
       {
146
            case 0:
147
                formattedString = "There is noone with that age.\n";
148
                break;
149
150
           case 1:
151
               formattedString = "there is a grand total of one person with "
152
                                  "that specific age.";
153
                break;
154
155
            default:
156
                formattedString = "There are a total of ";
157
                formattedString += to_string(totalOccurrences);
158
               formattedString += " people with that age.";
159
                // do something
160
                break;
161
       }
162
163
      cout << formattedString << endl;</pre>
164
165
166
      cout << PROMPT NAME;
       getline(cin, searchString);
167
168
        nameIndex = FindString(NAMES, AR LEN, searchString);
169
170
        formattedString = "";
171
172
      if (nameIndex == AR LEN)
173
174
            formattedString = "Im sorry, \"";
175
            formattedString += searchString += "\" was not found in my records.";
176
177
       else
178
179
            formattedString += NAMES[nameIndex];
180
            formattedString += "is";
            formattedString += to_string(AGES[nameIndex]) += " years old and "
181
                                                             "exists at index #";
182
183
            formattedString += to string(nameIndex);
```

```
184
      }
185
186
       cout << formattedString << endl << endl;</pre>
187
188
       // fetch the index of the oldest person
       oldestPersonIndex = FindLargestInt(AGES, AR LEN);
189
190
       // and assign their name to a value we can more easily use
191
       oldestPerson = NAMES[oldestPersonIndex];
192
193
       // fetch the index of the youngest person
194
       youngestPersonIndex = FindSmallestInt(AGES, AR LEN);
       youngestPerson = NAMES[youngestPersonIndex];
195
196
      cout << "The oldest person is " << oldestPerson << " who is "</pre>
197
198
            << AGES[oldestPersonIndex] << " years old which exists at index #"
199
            << oldestPersonIndex << endl;
200
201
       cout << "The youngest person is " << youngestPerson << " who is "</pre>
202
             << AGES[youngestPersonIndex] << " years old which exists at index #"
203
             << oldestPersonIndex << endl;
204
205
       sumAges = SumIntArray(AGES, AR LEN);
      cout << "Overall, the total combined age is " << sumAges << endl;</pre>
206
207
       cout << left << left;</pre>
208
       209
       cout << "==-- Begin Part 3&4 --==" << endl;
210
211
       212
      cout << " ----- Test #1=---- << endl;
213
214
215
      cout << fixed << setprecision(2);</pre>
216
       // compute average
217
      averageAges = AverageIntArray(AGES, AR LEN);
218
219
      cout << "The average of the array is " << averageAges << endl;</pre>
220
      // restore defaults
221
       cout << scientific << setprecision(6);</pre>
222
223
       cout << " ----- Test #2=---- << endl;
224
       // loop over caluclating first instance of the array
225
       for (int i = 0; i < 4; ++i)
226
      {
227
           int myInt;
           int result;
228
229
           myInt = SaneInputCinInt(0, 100, PROMPT FIND INT);
230
231
           result = FindFirstInstance(AGES, AR LEN, myInt);
232
233
           if (result == AR LEN)
234
               cout << myInt << " does not exist within the array." << endl;</pre>
235
236
            }
237
           else
238
           {
239
               cout << NAMES[result] << " is " << AGES[result] << " years old"</pre>
                    << "(index # " << result << ')' << endl;
240
241
242
           }
243
       }
244
```

```
245
      cout << endl;
246
       cout << " ----- Test #3=---- << endl;
247
248
       for (int i = 0; i < 4; ++i)
249
250
            int instances;
251
            int input;
252
253
            input = SaneInputCinInt(0, 100, PROMPT FIND INT);
254
            instances = FindOccurrencesInt(AGES, AR LEN, input);
255
256
            cout << input << " Occured " << instances << " time(s)." << endl;</pre>
       }
257
258
259
      cout << endl;
      cout << " ----- Test #4=---- << endl;
260
261
       // test for name search
262
       for (int i = 0; i < 4; ++i)
263
       {
264
            int resultIndex;
265
            string searchName;
266
267
            cout << PROMPT NAME;</pre>
            // fetch a name
268
269
            getline(cin, searchName);
270
271
            // get the result
272
            resultIndex = FindString(NAMES, AR LEN, searchName);
273
274
            if (resultIndex != AR LEN)
275
276
                // array hit
277
                cout << searchName << " is " << AGES[resultIndex] << " years old"</pre>
278
                     << "( index # " << resultIndex << ')';
279
            }
280
            else
281
282
                // array miss
283
                cout << searchName << " does not exist within the array.";</pre>
284
285
            cout << endl;</pre>
286
287
      cout << " ----- Test #5=---- << endl << endl;
288
289
        {
290
            int youngestIndex;
291
            youngestIndex = FindSmallestInt(AGES, AR LEN);
292
293
294
            cout << "The youngest person is " << NAMES[youngestIndex]</pre>
                 << " who is ";
295
            cout << AGES[youngestIndex] << " years old (Index # "</pre>
296
297
                 << youngestIndex << ')';
298
299
        cout << endl;</pre>
300
        cout << " ----- Test #6=---- << endl << endl;
301
302
            int oldestIndex;
303
            oldestIndex = FindLargestInt(AGES, AR LEN);
304
305
            cout << "The oldest person is " << NAMES[oldestIndex]</pre>
```

```
306
308
            << oldestIndex << ')';
309
   cout << endl << " ----- Test #7=---- << endl << endl;
310
311
     {
        int mSumAges;
312
313
314
        mSumAges = SumIntArray(AGES, AR LEN);
315
        cout << "Sum of all elements of the array is: " << mSumAges;</pre>
316
     }
     cout << endl;</pre>
317
    cout << "Done.";
318
319
320
     return 0;
321 }
322
323
```