```
 1: #include <iostream>
 2: #include <fstream>
 3: #include <string>
 4: using namespace std;
 5:
 6: void bigger(string*& words, int*& count, int& cap, int size);
 7: int findWord(string* words, int size, const string& w);
 8: void print(string* words, int* counts, int size);
 9:
10:
11:
12: int main ()
13: {
14:     int cap = 2; // start with cap at 2
15:     int size = 0; // start w/0 unique words
16:     string* words = new string[cap];
17:     int* count = new int[cap]();
18:
19:     string w = "";
20:
21:     ifstream fin ("big_doc.txt");
22:     if (!fin)
23:     {
24:         cout << "Unable to open file";
25:         return EXIT_FAILURE;
26:     }
27:
28:     while (fin >> w)
29:     {
30:         int idx = findWord(words, size, w);
31:
32:         if (idx == -1)
33:         {
34:             if (size == cap)
35:             {
36:                 bigger(words, count, cap, size);
37:             }
38:
39:             words[size] = w;
40:             count[size] = 1;
41:             size++;
42:         }
43:         else
44:         {
45:             count[idx]++;
46:         }
47:     }
48:
49:     print(words, count, size);
50:
51:     delete[] words;
52:     delete[] count;
53:
54:     return 0;
55: }
56:
57:
58:
59: void bigger(string*& words, int*& count, int& cap, int size)
60: {
```

```
 61:     cap *= 2;
 62:
 63:     string* newWords = new string[cap];
 64:     int* newCount = new int[cap]();
 65:
 66:     for (int i = 0; i < size; i++)
 67:     {
 68:         newWords[i] = words[i];
 69:         newCount[i] = count[i];
 70:     }
 71:
 72:     delete[] words;
 73:     delete[] count;
 74:
 75:     words = newWords;
 76:     count = newCount;
 77: }
 78:
 79: int findWord(string* words, int size, const string& w)
 80: {
 81:     for(int i = 0; i < size; i++)
 82:     {
 83:         if (words[i] == w)
 84:         {
 85:             return i;
 86:         }
 87:     }
 88:     return -1;
 89: }
 90:
 91: void print(string* words, int* counts, int size)
 92: {
 93:     if (size == 0)
 94:     {
 95:         cout << "No words to display." << endl;
 96:         return;
 97:     }
 98:
 99:     int limit = 10;
100:     if (size < 10)
101:     {
102:         limit = size;
103:     }
104:
105:     for (int i = 0; i < limit; i++)
106:     {
107:         cout << words[i] << " " << counts[i] << endl;
108:     }
109:
110:     for (int i = 10; i < size; i += 15)
111:     {
112:         cout << words[i] << " " << counts[i] << endl;
113:     }
114:
115:     //min and max
116:     int maxIdx = 0;
117:     int minIdx = 0;
118:
119:     for (int i = 0; i < size; i++)
120:     {
```

```
121:         if (counts[i] > counts[maxIdx])
122:         {
123:             maxIdx = i;
124:         }
125:         if (counts[i] < counts[minIdx])
126:         {
127:             minIdx = i;
128:         }
129:     }
130:
131:     cout << "Most frequent word count: " << counts[maxIdx] << endl;
132:     for (int i = 0; i < size; i++)
133:     {
134:         if (counts[i] == counts[maxIdx])
135:         {
136:             cout << words[i] << endl;
137:         }
138:     }
139:
140:     cout << "Least frequent word count: " << counts[minIdx] << endl;
141:     int printed = 0;
142:     for (int i = 0; i < size && printed < 10; i++)
143:     {
144:         if (counts[i] == counts[minIdx])
145:         {
146:             cout << words[i] << endl;
147:             printed++;
148:         }
149:     }
150:
151: }
152:
153:
154: /*
155: text 33
156: processing 15
157: is 26
158: an 4
159: enduring 1
160: task 2
161: in 21
162: computer 1
163: science 1
164: it 18
165: not 5
166: data 9
167: medical 3
168: both 1
169: character 3
170: once 1
171: programmers 3
172: natural 2
173: by 3
174: reliable 1
175: teaching 1
176: clean 1
177: arrays 2
178: major 1
179: be 7
180: files 3
```

```
181: done 1
182: millions 1
183: concepts 1
184: large 4
185: diagnosis 1
186: compiler 2
187: here 1
188: beyond 2
189: finding 1
190: foundation 2
191: principles 2
192: seed 1
193: happens 1
194: terminators 1
195: ambiguity 1
196: gracefully 1
197: workflows 1
198: report 1
199: between 2
200: reading 1
201: while 1
202: Most frequent word count: 63
203: and
204: Least frequent word count: 1
205: enduring
206: computer
207: science
208: side
209: topic
210: element
211: nearly
212: digital
213: people
214: organizations
215:
216: --------------------------------
217: Process exited after 0.2348 seconds with return value 0
218: Press any key to continue . . .
219: */
```