

```
1 #include <iostream>
2 #include <fstream>
3 #include <string>
4 #include "stack.h"
5 #include "queues.h"
6 #include <cstring>
7 using namespace std;
8
9 int main()
10 {
11     fstream fin;
12     fstream sent1;
13     fstream sent2;
14     fstream sent3;
15     fstream sent4;
16     fstream sent5;
17     fstream sent6;
18     fstream sent7;
19     ofstream fout;
20     fin.open("palindrome_input_file.txt");
21     if (!(fin.is_open()))
22     {
23         cout << "The input file did not open" << endl;
24     }
25     sent1.open("palindrome_input_filearray.txt");
26     if (!(sent1.is_open()))
27     {
28         cout << "The input file did not open" << endl;
29     }
30     sent2.open("palindrome_sentence2.txt");
31     if (!(sent2.is_open()))
32     {
33         cout << "The input file did not open" << endl;
34     }
35     sent3.open("palindrome_sentence3.txt");
36     if (!(sent3.is_open()))
37     {
38         cout << "The input file did not open" << endl;
39     }
40     sent4.open("palindrome_sentence4.txt");
41     if (!(sent4.is_open()))
42     {
43         cout << "The input file did not open" << endl;
44     }
45     sent5.open("palindrome_sentence5.txt");
46     if (!(sent5.is_open()))
47     {
48         cout << "The input file did not open" << endl;
49     }
```

```
50     sent6.open("palindrome_sentence6.txt");
51     if (!(sent6.is_open()))
52     {
53         cout << "The input file did not open" << endl;
54     }
55     sent7.open("palindrome_sentence7.txt");
56     if (!(sent7.is_open()))
57     {
58         cout << "The input file did not open" << endl;
59     }
60
61     fout.open("Output_file.txt");
62
63     // this varriable will be used to input all our characters and pass into methods.
64     char str1;
65     char str2;
66     char str3;
67     char str4;
68     char str5;
69     char str6;
70     char str7;
71     // these varriable will capture the char that is deleted in stack.
72     char sents;
73     char sents2;
74     char sents3;
75     char sents4;
76     char sents5;
77     char sents6;
78     char sents7;
79     // this varriable will capture the cahr that is deleted in queue.
80     char sentq;
81     char sentq2;
82     char sentq3;
83     char sentq4;
84     char sentq5;
85     char sentq6;
86     char sentq7;
87     // this will be our true false operator for our output of the palindrome.
88     bool palindrome = true;
89     // the number stacks are our char arrays that we will use to compare to se if its a palindrome.
90     stack<char> one;
91     stack<char> two;
92     stack<char> three;
93     stack<char> four;
94     stack<char> five;
95     stack<char> six;
```

```
96     stack<char> seven;
97     // the number q's are our char arrays that we will use to compare to
    see if its a palindrome.
98     queues<char> oneq;
99     queues<char> twoq;
100    queues<char> threeq;
101    queues<char> fourq;
102    queues<char> fiveq;
103    queues<char> sixq;
104    queues<char> sevenq;
105    /* These are all our temp queues. Had to make multiple because whenever
    I deleted the queue fully and tried to enter
106    more back in for the second sentence. The characters would not save
    inside the linked list to print.
107    */
108    queues<char>temp;
109    queues<char>temp2;
110    queues<char>temp3;
111    queues<char>temp4;
112    queues<char>temp5;
113    queues<char>temp6;
114    queues<char>temp7;
115    // The orgs are the origianl sentences with spaces. We use them for
    output purposes
116    queues<char>org;
117    queues<char>org2;
118    queues<char>org3;
119    queues<char>org4;
120    queues<char>org5;
121    queues<char>org6;
122    queues<char>org7;
123
124    for (int i = 0; i < 19; i++)
125    {
126        sent1.get(str1);
127        if (!(one.isfull()))
128        {
129            one.push(str1);
130        }
131        if (!(oneq.isfull()))
132        {
133            oneq.enqueue(str1);
134        }
135    }
136
137    while (!(one.isEmpty()))
138    {
139        sents = one.pop();
140        sentq = oneq.delqueue();
```

```
141         if (!(temp.isfull()))
142         {
143             temp.enqueue(sentq);
144         }
145         if ((char)toupper(sents) != (char)toupper(sentq))
146         {
147             palindrome = false;
148         }
149         else
150             palindrome = true;
151     }
152 }
153
154 for (int i = 0; i < 25; i++)
155 {
156     fin.get(str1);
157     if (!(org.isfull()))
158     {
159         org.enqueue(str1);
160     }
161 }
162
163 org.print(fout);
164
165 if (palindrome == true)
166 {
167     fout << " is a palindrome";
168 }
169 else
170 {
171     fout << " is not a palindrome";
172 }
173
174 // This is start of second sentence
175
176 for (int i = 0; i < 12; i++)
177 {
178     sent2.get(str2);
179     if (!(two.isfull()))
180     {
181         two.push(str2);
182     }
183     if (!(twoq.isfull()))
184     {
185         twoq.enqueue(str2);
186     }
187 }
188
189
```

```
190     while (!(two.isempty()))
191     {
192         sents2 = two.pop();
193         sentq2 = twoq.delqueue();
194         if (!(temp2.isfull()))
195         {
196             temp2.enqueue(sentq2);
197         }
198         if ((char)toupper(sents2) != (char)toupper(sentq2))
199         {
200             palindrome = false;
201         }
202         else
203             palindrome = true;
204     }
205
206     for (int i = 0; i < 16; i++)
207     {
208         fin.get(str2);
209         if (!(org2.isfull()))
210         {
211             org2.enqueue(str2);
212         }
213     }
214
215     org2.print(fout);
216
217     if (palindrome == true)
218     {
219         fout << " is a palindrome";
220     }
221     else
222     {
223         fout << " is not a palindrome";
224     }
225
226     // This is the third sentence
227
228     for (int i = 0; i < 24; i++)
229     {
230         sent3.get(str3);
231         if (!(three.isfull()))
232         {
233             three.push(str3);
234         }
235         if (!(threeq.isfull()))
236         {
237             threeq.enqueue(str3);
238         }
```

```
239     }
240 }
241
242
243 while (!(three.isempty()))
244 {
245     sents3 = three.pop();
246     sentq3 = threeq.dequeue();
247     if (!(temp3.isfull()))
248     {
249         temp3.enqueue(sentq3);
250     }
251     if ((char)toupper(sents3) != (char)toupper(sentq3))
252     {
253         palindrome = false;
254     }
255     else
256         palindrome = true;
257 }
258
259 for (int i = 0; i < 29; i++)
260 {
261     fin.get(str3);
262     if (!(org3.isfull()))
263     {
264         org3.enqueue(str3);
265     }
266 }
267
268
269 org3.print(fout);
270
271 if (palindrome == true)
272 {
273     fout << " is a palindrome";
274 }
275 else
276 {
277     fout << " is not a palindrome";
278 }
279
280 // This is fourth sentence
281
282 for (int i = 0; i < 7; i++)
283 {
284     sent4.get(str4);
285     if (!(four.isfull()))
286     {
287         four.push(str4);
```

```
288     }
289     if (!(fourq.isfull()))
290     {
291         fourq.enqueue(str4);
292     }
293 }
294
295
296
297 while (!(four.isempty()))
298 {
299     sents4 = four.pop();
300     sentq4 = fourq.delqueue();
301     if (!(temp4.isfull()))
302     {
303         temp4.enqueue(sentq4);
304     }
305     if ((char)toupper(sents4) != (char)toupper(sentq4))
306     {
307         palindrome = false;
308     }
309     else
310         palindrome = true;
311 }
312
313
314 for (int i = 0; i < 9; i++)
315 {
316     fin.get(str4);
317     if (!(org4.isfull()))
318     {
319         org4.enqueue(str4);
320     }
321 }
322
323 org4.print(fout);
324
325 if (palindrome == true)
326 {
327     fout << " is a palindrome";
328 }
329 else
330 {
331     fout << " is not a palindrome";
332 }
333
334 // This is the fifth sentence
335
336 for (int i = 0; i < 21; i++)
```

```
337     {
338         sent5.get(str5);
339         if (!(five.isfull()))
340         {
341             five.push(str5);
342         }
343         if (!(fiveq.isfull()))
344         {
345             fiveq.enqueue(str5);
346         }
347     }
348
349
350     while (!(five.isempty()))
351     {
352         sents5 = five.pop();
353         sentq5 = fiveq.delqueue();
354         if (!(temp5.isfull()))
355         {
356             temp5.enqueue(sentq5);
357         }
358         if ((char)toupper(sents5) != (char)toupper(sentq5))
359         {
360             palindrome = false;
361         }
362         else
363             palindrome = true;
364     }
365
366
367     for (int i = 0; i < 28; i++)
368     {
369         fin.get(str5);
370         if (!(org5.isfull()))
371         {
372             org5.enqueue(str5);
373         }
374     }
375
376     org5.print(fout);
377
378     if (palindrome == true)
379     {
380         fout << " is a palindrome";
381     }
382     else
383     {
384         fout << " is not a palindrome";
385     }
```



```
386
387     // This is sentence 6
388
389     for (int i = 0; i < 11; i++)
390     {
391         sent6.get(str6);
392         if (!(six.isfull()))
393         {
394             six.push(str6);
395         }
396         if (!(sixq.isfull()))
397         {
398             sixq.enqueue(str6);
399         }
400     }
401
402
403     while (!(six.isEmpty()))
404     {
405         sents6 = six.pop();
406         sentq6 = sixq.delqueue();
407         if (!(temp6.isfull()))
408         {
409             temp6.enqueue(sentq6);
410         }
411         if ((char)toupper(sents6) != (char)toupper(sentq6))
412         {
413             palindrome = false;
414         }
415         else
416             palindrome = true;
417     }
418
419
420     for (int i = 0; i < 15; i++)
421     {
422         fin.get(str6);
423         if (!(org6.isfull()))
424         {
425             org6.enqueue(str6);
426         }
427     }
428
429     org6.print(fout);
430
431     if (palindrome == true)
432     {
433         fout << " is a palindrome";
434     }
```

```
435     else
436     {
437         fout << " is not a palindrome";
438     }
439
440     // This is the seventh sentence
441
442     for (int i = 0; i < 12; i++)
443     {
444         sent7.get(str7);
445         if (!(seven.isfull()))
446         {
447             seven.push(str7);
448         }
449         if (!(sevenq.isfull()))
450         {
451             sevenq.enqueue(str7);
452         }
453     }
454
455
456     while (!(seven.isEmpty()))
457     {
458         sents7 = seven.pop();
459         sentq7 = sevenq.delqueue();
460         if (!(temp7.isfull()))
461         {
462             temp7.enqueue(sentq7);
463         }
464         if ((char)toupper(sents7) != (char)toupper(sentq7))
465         {
466             palindrome = false;
467         }
468         else
469             palindrome = true;
470     }
471
472
473     for (int i = 0; i < 14; i++)
474     {
475         fin.get(str7);
476         if (!(org7.isfull()))
477         {
478             org7.enqueue(str7);
479         }
480     }
481
482     org7.print(fout);
483
```

```
484     if (palindrome == true)
485     {
486         fout << " is a palindrome";
487     }
488     else
489     {
490         fout << " is not a palindrome";
491     }
492     return 0;
493 }
```