

```
1 *****
2 * PROGRAMMED BY : Nick Reardon
3 * CLASS        : CS1D
4 * SECTION      : MW - 2:30p
5 * Assignment #3 : Stacks, Queues, Dequeus
6 *****
7
8             Assignment #3 - Stacks, Queues, Dequeus
9
10 Given the following data:
11
12 Input for the string stacks/queues/deques
13 Mark, Alan, Jennifer, Jordyn, Eric, JoAnn, Bryan
14
15 Input for the double stacks/queues/deques
16 2019.1, 44.44, 888.55, 200.12, 123.123, 8.445,
17
18 A. Implement and print (top of stack to bottom) the stacks
19    using the STL <stack> with the above data.
20 B. Delete Jordyn and 200.12 from the above stacks (you will
21    need to delete others) using the STL <stack> and print (top of
22    stack to bottom) the remaining elements in the stacks.
23 C. Implement and print (top of stack to bottom) the stacks
24    using a singly linked list using the above data. Do not use the
25    STL.
26 D. Delete Jordyn and 200.12 from the above stacks (you will
27    need to delete others) and print (top of stack to bottom) the
28    remaining elements in the stacks. Do not use the STL.
29 E. Implement and print the queues using either a circular array
30    or a linked list using the above data. Do not use the STL.
31 F. Delete JoAnn and 200.12 from the above queues (you will
32    need to delete others) and print the remaining elements in
33    the queues. Do not use the STL.
34 G. Implement and print the dequeus using a linked list using the
35    above data (using push front). Do not use the STL.
36 H. Delete JoAnn (pop front) and 200.12 (pop back) from the
37    above dequeus (you will need to delete others) and print the
38    remaining elements in the dequeus. Do not use the STL.
39
40 Label your output (part A, part B, part C, etc.)
41 Do not put deleted elements back on the data structures.
42
43 I. Implement the Parentheses Algorithm without using the
44    STL). Test your algorithm with the following mathematical
45    statements.
46    a.  $(12x + 6) (2x - 4)$ 
47    b.  $\{2x + 5\} (6x+4)$ 
48    c.  $\{2x + 7\} (12x + 6)$ 
49    d.  $\{\{8x+5\} - 5x[9x+3]\}$ 
50    e.  $((4x+8) - x[4x+3]))$ 
51    f.  $[(5x - 5) - 4x[6x + 2]]$ 
52    g.  $\{(8x+5) - 6x[9x+3]\}$ 
```

```
53
54 J. (extra credit ù 3 points) If valid, write software to evaluate
55     the valid expressions above assuming x = -2.
56
57 Your output should CLEARLY demonstrate the above. Print out
58 the part number before you display the stacks/queues/deques.
59
60 Due on February 3rd
61
62
63 *****
64
65 Reading from file into string stacks/queue/deque
66 Reading from file into double stacks/queue/deque
67
68     --- PART A ---
69
70 Printing STL stack:
71 Bryan
72 JoAnn
73 Eric
74 Jordyn
75 Jennifer
76 Alan
77 Mark
78
79 Printing STL stack:
80 8.445
81 123.123
82 200.12
83 888.55
84 44.44
85 2019.1
86
87
88     --- PART B ---
89
90 Deleting Jordyn from the STL stack
91 deleting Bryan
92 deleting JoAnn
93 deleting Eric
94 deleting Jordyn
95
96 Deleting 200.12 from the STL stack
97 deleting 8.445
98 deleting 123.123
99 deleting 200.12
100
101 Printing STL stack:
102 Jennifer
103 Alan
104 Mark
```

```
105
106 Printing STL stack:
107 888.55
108 44.44
109 2019.1
110
111
112 --- PART C ---
113
114 Printing singly linked list stacks:
115
116 Printing stack:
117 Bryan
118 JoAnn
119 Eric
120 Jordyn
121 Jennifer
122 Alan
123 Mark
124
125 Printing stack:
126 8.445
127 123.123
128 200.12
129 888.55
130 44.44
131 2019.1
132
133
134 --- PART D ---
135
136 Deleting Jordyn from the linked list stack
137 Popped item is Bryan
138 Popped item is JoAnn
139 Popped item is Eric
140 Popped item is Jordyn
141
142 Deleting 200.12 from the linked list stack
143 Popped item is 8.445
144 Popped item is 123.123
145 Popped item is 200.12
146
147 Printing stack:
148 Jennifer
149 Alan
150 Mark
151
152 Printing stack:
153 888.55
154 44.44
155 2019.1
156
```

```
157
158     --- PART E ---
159
160 Printing singly linked list queues:
161
162 Printing queue:
163 2019.1
164 44.44
165 888.55
166 200.12
167 123.123
168 8.445
169
170 Printing queue:
171 Mark
172 Alan
173 Jennifer
174 Jordyn
175 Eric
176 JoAnn
177 Bryan
178
179
180     --- PART F ---
181
182 Deleting Jordyn from the linked list queue
183 deQueued item is Mark
184 deQueued item is Alan
185 deQueued item is Jennifer
186 deQueued item is Jordyn
187
188 Deleting 200.12 from the linked list queue
189 deQueued item is 2019.1
190 deQueued item is 44.44
191 deQueued item is 888.55
192 deQueued item is 200.12
193
194 Printing queue:
195 123.123
196 8.445
197
198 Printing queue:
199 Eric
200 JoAnn
201 Bryan
202
203
204     --- PART G ---
205
206 Printing doubly linked list dequeues:
207
208 Printing deque front to back:
```

```
209 Bryan
210 JoAnn
211 Eric
212 Jordyn
213 Jennifer
214 Alan
215 Mark
216
217 Printing deque front to back:
218 8.445
219 123.123
220 200.12
221 888.55
222 44.44
223 2019.1
224
225
226 --- PART H ---
227
228 Deleting Jordyn, using pop front, from the doubly linked list deque
229 removing from front: Bryan
230 removing from front: JoAnn
231 removing from front: Eric
232 removing from front: Jordyn
233
234 Deleting 200.12, using pop back, from the doubly linked list deque
235 removing from back: 2019.1
236 removing from back: 44.44
237 removing from back: 888.55
238 removing from back: 200.12
239
240 Printing deque front to back:
241 Jennifer
242 Alan
243 Mark
244
245 Printing deque front to back:
246 8.445
247 123.123
248
249
250 --- PART I ---
251
252 Testing Parentheses Algorithm - using singly linked list stack
253
254
255 (12x + 6) (2x - 4)
256 Balanced
257
258 {2x + 5} (6x+4)
259 Balanced
260
```

```
261 {2x + 7) (12x + 6)
262 Not Balanced
263
264 {{8x+5) - 5x[9x+3]}})
265 Not Balanced
266
267 (((4x+8) - x[4x+3]))))
268 Not Balanced
269
270 [(5x - 5) - 4x[6x + 2]]
271 Balanced
272
273 {(8x+5) - 6x[9x+3]]
274 Not Balanced
275 Press any key to continue . . .
```