

```
1 *****
2 * PROGRAMMED BY   : Nick Reardon
3 * CLASS           : CS1D
4 * SECTION         : MW - 2:30p
5 * Extra Credit #1 : Extendable Array
6 *****
7
8             Extra Credit #1 - Extendable Array
9
10    Implement the vector ADT by means of an extendable array
11 (expands dynamically) used in a circular fashion, so that
12 insertions and deletions at the beginning and end run in constant
13 time. Print the circular array before and after each insertion and
14 deletion. Insertions and deletion that are not at the beginning or
15 end will run in O(n) time. Test the vector ADT methods (inserts,
16 deletes, size, isEmpty, etc.). The initial array size is 4. Print out
17 the index (rank) of the array with the corresponding value.
18
19 Do not use the <vector> STL.
20
21    Use the following data:
22
23    Insert 14, 5, 4, 1, 21, 33, 12, 8
24    Delete 14
25    Delete 1
26    Insert 9 at the front of the vector
27    Insert 7 after 33
28    Insert 44 at the end of the vector
29
30 Due February 12
31
32 *****
33
34 Initializing int ExtendableArray vector with capacity of 4
35
36
37 Calling print method to show the vector is empty
38 Exception - Error number 2:    Array is empty - nothing to print
39
40 Inserting 14, 5, 4, 1 into vector using insertBack()
41
42 Current vector
43 [ 0 ] = 14    <- First index
44 [ 1 ] = -842150451
45 [ 2 ] = -842150451
46 [ 3 ] = -842150451
47 Useable order: 14
48
49 Current vector
50 [ 0 ] = 14    <- First index
51 [ 1 ] = 5     <- Last index
52 [ 2 ] = -842150451
```

```
53 [ 3 ] = -842150451
54 Useable order: 14, 5
55
56 Current vector
57 [ 0 ] = 14   <- First index
58 [ 1 ] = 5
59 [ 2 ] = 4   <- Last index
60 [ 3 ] = -842150451
61 Useable order: 14, 5, 4
62
63 Current vector
64 [ 0 ] = 14   <- First index
65 [ 1 ] = 5
66 [ 2 ] = 4
67 [ 3 ] = 1   <- Last index
68 Useable order: 14, 5, 4, 1
69
70
71 Inserting 21 into vector using insertBack()
72
73 Insertion would exceed capacity, so protected expand()
74 method is called to double the vector's capacity
75
76 Current vector
77 [ 0 ] = 14   <- First index
78 [ 1 ] = 5
79 [ 2 ] = 4
80 [ 3 ] = 1
81 [ 4 ] = 21   <- Last index
82 [ 5 ] = -842150451
83 [ 6 ] = -842150451
84 [ 7 ] = -842150451
85 Useable order: 14, 5, 4, 1, 21
86
87 Inserting 33, 12, 8 intop vector using insertBack()
88 Useable order: 14, 5, 4, 1, 21
89
90 Current vector
91 [ 0 ] = 14   <- First index
92 [ 1 ] = 5
93 [ 2 ] = 4
94 [ 3 ] = 1
95 [ 4 ] = 21
96 [ 5 ] = 33   <- Last index
97 [ 6 ] = -842150451
98 [ 7 ] = -842150451
99 Useable order: 14, 5, 4, 1, 21, 33
100
101 Current vector
102 [ 0 ] = 14   <- First index
103 [ 1 ] = 5
104 [ 2 ] = 4
```

```
105 [ 3 ] = 1
106 [ 4 ] = 21
107 [ 5 ] = 33
108 [ 6 ] = 12    <- Last index
109 [ 7 ] = -842150451
110 Useable order: 14, 5, 4, 1, 21, 33, 12
111
112 Current vector
113 [ 0 ] = 14    <- First index
114 [ 1 ] = 5
115 [ 2 ] = 4
116 [ 3 ] = 1
117 [ 4 ] = 21
118 [ 5 ] = 33
119 [ 6 ] = 12
120 [ 7 ] = 8    <- Last index
121 Useable order: 14, 5, 4, 1, 21, 33, 12, 8
122
123
124 removing 14, using eraseFront()
125
126 Current vector
127 [ 0 ] = 14
128 [ 1 ] = 5    <- First index
129 [ 2 ] = 4
130 [ 3 ] = 1
131 [ 4 ] = 21
132 [ 5 ] = 33
133 [ 6 ] = 12
134 [ 7 ] = 8    <- Last index
135 Useable order: 5, 4, 1, 21, 33, 12, 8
136
137
138 removing 1, using eraseAt()
139
140 Current vector
141 [ 0 ] = 14
142 [ 1 ] = 5
143 [ 2 ] = 5    <- First index
144 [ 3 ] = 4
145 [ 4 ] = 21
146 [ 5 ] = 33
147 [ 6 ] = 12
148 [ 7 ] = 8    <- Last index
149 Useable order: 5, 4, 21, 33, 12, 8
150
151
152 Inserting 7 after 33 into vector using insertAt()
153
154 Current vector
155 [ 0 ] = 8    <- Last index
156 [ 1 ] = 5
```

```
157 [ 2 ] = 5    <- First index
158 [ 3 ] = 4
159 [ 4 ] = 21
160 [ 5 ] = 33
161 [ 6 ] = 12
162 [ 7 ] = 7
163 Useable order: 5, 4, 21, 33, 12, 7, 8
164
165
166 Inserting 44 into vector using insertBack()
167
168 Current vector
169 [ 0 ] = 8
170 [ 1 ] = 44    <- Last index
171 [ 2 ] = 5     <- First index
172 [ 3 ] = 4
173 [ 4 ] = 21
174 [ 5 ] = 33
175 [ 6 ] = 12
176 [ 7 ] = 7
177 Useable order: 5, 4, 21, 33, 12, 7, 8, 44
178 Press any key to continue . . .
```