

```
1 Running tests for sequence class with a dynamic array
2
3 START OF TEST 1:
4 Testing insert, attach, and the constant member functions (4 points).
5 Starting with an empty sequence.
6 Testing that size() returns 0 ... Passed.
7 Testing that is_item() returns false ... Passed.
8 I'll call start() and look at the items one more time...
9 All tests passed for this sequence.
10
11 I am now using attach to put 10 into an empty sequence.
12 Testing that size() returns 1 ... Passed.
13 Testing that is_item() returns true ... Passed.
14 The cursor should be at item [0] of the sequence
15 (counting the first item as [0]). I will advance the cursor
16 to the end of the sequence, checking that each item is correct...Passed.
17 I'll call start() and look at the items one more time...
18 The cursor should be at item [0] of the sequence
19 (counting the first item as [0]). I will advance the cursor
20 to the end of the sequence, checking that each item is correct...Passed.
21 All tests passed for this sequence.
22
23 I am now using insert to put 10 into an empty sequence.
24 Testing that size() returns 1 ... Passed.
25 Testing that is_item() returns true ... Passed.
26 The cursor should be at item [0] of the sequence
27 (counting the first item as [0]). I will advance the cursor
28 to the end of the sequence, checking that each item is correct...Passed.
29 I'll call start() and look at the items one more time...
30 The cursor should be at item [0] of the sequence
31 (counting the first item as [0]). I will advance the cursor
32 to the end of the sequence, checking that each item is correct...Passed.
33 All tests passed for this sequence.
34
35 I am now using attach to put 10,20,30 in an empty sequence.
36 Then I move the cursor to the start and insert 5.
37 Testing that size() returns 4 ... Passed.
38 Testing that is_item() returns true ... Passed.
39 The cursor should be at item [0] of the sequence
40 (counting the first item as [0]). I will advance the cursor
41 to the end of the sequence, checking that each item is correct...Passed.
42 I'll call start() and look at the items one more time...
43 The cursor should be at item [0] of the sequence
44 (counting the first item as [0]). I will advance the cursor
45 to the end of the sequence, checking that each item is correct...Passed.
46 All tests passed for this sequence.
47
48 I am now using attach to put 10,20,30 in an empty sequence.
49 Then I move the cursor to the start, advance once, and insert 15.
50 Testing that size() returns 4 ... Passed.
51 Testing that is_item() returns true ... Passed.
52 The cursor should be at item [1] of the sequence
53 (counting the first item as [0]). I will advance the cursor
54 to the end of the sequence, checking that each item is correct...Passed.
55 I'll call start() and look at the items one more time...
56 The cursor should be at item [0] of the sequence
57 (counting the first item as [0]). I will advance the cursor
58 to the end of the sequence, checking that each item is correct...Passed.
59 All tests passed for this sequence.
60
61 I am now using attach to put 10,20,30 in an empty sequence.
62 Then I move the cursor to the start and attach 15 after the 10.
63 Testing that size() returns 4 ... Passed.
64 Testing that is_item() returns true ... Passed.
65 The cursor should be at item [1] of the sequence
66 (counting the first item as [0]). I will advance the cursor
67 to the end of the sequence, checking that each item is correct...Passed.
68 I'll call start() and look at the items one more time...
69 The cursor should be at item [0] of the sequence
```

```
70 (counting the first item as [0]). I will advance the cursor
71 to the end of the sequence, checking that each item is correct...Passed.
72 All tests passed for this sequence.
73
74 All tests of this first function have been passed.
75 Test 1 got 4 points out of a possible 4.
76 END OF TEST 1.
77
78
79 START OF TEST 2:
80 Testing situations where the cursor goes off the sequence (4 points).
81 Using attach to put 20 and 30 in the sequence, and then calling
82 advance, so that is_item should return false ... passed.
83 Inserting 10, which should go at the sequence's front.
84 Then calling advance three times to run cursor off the sequence ... passed.
85 Calling attach to put the numbers 40, 50, 60 ...300 at the sequence's end.
86 Now I will test that the sequence has 10, 20, 30, ...300.
87 All tests of this second function have been passed.
88 Test 2 got 4 points out of a possible 4.
89 END OF TEST 2.
90
91
92 START OF TEST 3:
93 Testing remove_current (4 points).
94 Using attach to build a sequence with 10,30.
95 Insert a 20 before the 30, so entire sequence is 10,20,30.
96 Testing that size() returns 3 ... Passed.
97 Testing that is_item() returns true ... Passed.
98 The cursor should be at item [1] of the sequence
99 (counting the first item as [0]). I will advance the cursor
100 to the end of the sequence, checking that each item is correct...Passed.
101 I'll call start() and look at the items one more time...
102 The cursor should be at item [0] of the sequence
103 (counting the first item as [0]). I will advance the cursor
104 to the end of the sequence, checking that each item is correct...Passed.
105 All tests passed for this sequence.
106
107 Remove the 20, so entire sequence is now 10,30.
108 Testing that size() returns 2 ... Passed.
109 Testing that is_item() returns true ... Passed.
110 The cursor should be at item [1] of the sequence
111 (counting the first item as [0]). I will advance the cursor
112 to the end of the sequence, checking that each item is correct...Passed.
113 I'll call start() and look at the items one more time...
114 The cursor should be at item [0] of the sequence
115 (counting the first item as [0]). I will advance the cursor
116 to the end of the sequence, checking that each item is correct...Passed.
117 All tests passed for this sequence.
118
119 Remove the 30, so entire sequence is now just 10 with no cursor.
120 Testing that size() returns 1 ... Passed.
121 Testing that is_item() returns false ... Passed.
122 I'll call start() and look at the items one more time...
123 All tests passed for this sequence.
124
125 Set the cursor to the start and remove the 10.
126 Testing that size() returns 0 ... Passed.
127 Testing that is_item() returns false ... Passed.
128 I'll call start() and look at the items one more time...
129 All tests passed for this sequence.
130
131 Using attach to build another sequence with 10,30.
132 Insert a 20 before the 30, so entire sequence is 10,20,30.
133 Testing that size() returns 3 ... Passed.
134 Testing that is_item() returns true ... Passed.
135 The cursor should be at item [1] of the sequence
136 (counting the first item as [0]). I will advance the cursor
137 to the end of the sequence, checking that each item is correct...Passed.
138 I'll call start() and look at the items one more time...
```

```
139 The cursor should be at item [0] of the sequence
140 (counting the first item as [0]). I will advance the cursor
141 to the end of the sequence, checking that each item is correct...Passed.
142 All tests passed for this sequence.
143
144 Remove the 20, so entire sequence is now 10,30.
145 Testing that size() returns 2 ... Passed.
146 Testing that is_item() returns true ... Passed.
147 The cursor should be at item [1] of the sequence
148 (counting the first item as [0]). I will advance the cursor
149 to the end of the sequence, checking that each item is correct...Passed.
150 I'll call start() and look at the items one more time...
151 The cursor should be at item [0] of the sequence
152 (counting the first item as [0]). I will advance the cursor
153 to the end of the sequence, checking that each item is correct...Passed.
154 All tests passed for this sequence.
155
156 Set the cursor to the start and remove the 10,
157 so the sequence should now contain just 30.
158 Testing that size() returns 1 ... Passed.
159 Testing that is_item() returns true ... Passed.
160 The cursor should be at item [0] of the sequence
161 (counting the first item as [0]). I will advance the cursor
162 to the end of the sequence, checking that each item is correct...Passed.
163 I'll call start() and look at the items one more time...
164 The cursor should be at item [0] of the sequence
165 (counting the first item as [0]). I will advance the cursor
166 to the end of the sequence, checking that each item is correct...Passed.
167 All tests passed for this sequence.
168
169 Remove the 30 from the sequence, resulting in an empty sequence.
170 Testing that size() returns 0 ... Passed.
171 Testing that is_item() returns false ... Passed.
172 I'll call start() and look at the items one more time...
173 All tests passed for this sequence.
174
175 Build a new sequence by inserting 30, 10, 20 (so the sequence
176 is 20, then 10, then 30). Then remove the 20.
177 Testing that size() returns 2 ... Passed.
178 Testing that is_item() returns true ... Passed.
179 The cursor should be at item [0] of the sequence
180 (counting the first item as [0]). I will advance the cursor
181 to the end of the sequence, checking that each item is correct...Passed.
182 I'll call start() and look at the items one more time...
183 The cursor should be at item [0] of the sequence
184 (counting the first item as [0]). I will advance the cursor
185 to the end of the sequence, checking that each item is correct...Passed.
186 All tests passed for this sequence.
187
188 Just for fun, I'll empty the sequence then fill it up, then
189 empty it again. During this process, I'll try to determine
190 whether any of the sequence's member functions access the
191 array outside of its legal indexes.
192 All tests of this third function have been passed.
193 Test 3 got 4 points out of a possible 4.
194 END OF TEST 3.
195
196
197 START OF TEST 4:
198 Testing the resize member function (2 points).
199 I will now resize a sequence to a larger capacity, and then
200 attach that many items. The sequence should NOT need to
201 resize itself under this situation.
202     Test passed.
203 Now I will call resize(1) for the sequence, but the actual
204 sequence should not change because the sequence already has
205 60 items.
206     Test passed.
207 All tests of this fourth function have been passed.
```

```
208 Test 4 got 2 points out of a possible 2.
209 END OF TEST 4.
210
211
212 START OF TEST 5:
213 Testing the copy constructor (2 points).
214 Copy constructor test: for an empty sequence.
215 Testing that size() returns 0 ... Passed.
216 Testing that is_item() returns false ... Passed.
217 I'll call start() and look at the items one more time...
218 All tests passed for this sequence.
219
220 Copy constructor test: for a sequence with cursor at tail.
221 Testing that size() returns 60 ... Passed.
222 Testing that is_item() returns true ... Passed.
223 The cursor should be at item [59] of the sequence
224 (counting the first item as [0]). I will advance the cursor
225 to the end of the sequence, checking that each item is correct...Passed.
226 I'll call start() and look at the items one more time...
227 The cursor should be at item [0] of the sequence
228 (counting the first item as [0]). I will advance the cursor
229 to the end of the sequence, checking that each item is correct...Passed.
230 All tests passed for this sequence.
231
232 Copy constructor test: for a sequence with cursor near middle.
233 Testing that size() returns 60 ... Passed.
234 Testing that is_item() returns true ... Passed.
235 The cursor should be at item [30] of the sequence
236 (counting the first item as [0]). I will advance the cursor
237 to the end of the sequence, checking that each item is correct...Passed.
238 I'll call start() and look at the items one more time...
239 The cursor should be at item [0] of the sequence
240 (counting the first item as [0]). I will advance the cursor
241 to the end of the sequence, checking that each item is correct...Passed.
242 All tests passed for this sequence.
243
244 Copy constructor test: for a sequence with cursor near middle.
245 Testing that size() returns 60 ... Passed.
246 Testing that is_item() returns true ... Passed.
247 The cursor should be at item [0] of the sequence
248 (counting the first item as [0]). I will advance the cursor
249 to the end of the sequence, checking that each item is correct...Passed.
250 I'll call start() and look at the items one more time...
251 The cursor should be at item [0] of the sequence
252 (counting the first item as [0]). I will advance the cursor
253 to the end of the sequence, checking that each item is correct...Passed.
254 All tests passed for this sequence.
255
256 Copy constructor test: for a sequence with no current item.
257 Testing that size() returns 60 ... Passed.
258 Testing that is_item() returns false ... Passed.
259 I'll call start() and look at the items one more time...
260 All tests passed for this sequence.
261
262 All tests of this fifth function have been passed.
263 Test 5 got 2 points out of a possible 2.
264 END OF TEST 5.
265
266
267 START OF TEST 6:
268 Testing the assignment operator (2 points).
269 Assignment operator test: for an empty sequence.
270 Testing that size() returns 0 ... Passed.
271 Testing that is_item() returns false ... Passed.
272 I'll call start() and look at the items one more time...
273 All tests passed for this sequence.
274
275 Assignment operator test: for a sequence with cursor at tail.
276 Testing that size() returns 60 ... Passed.
```

```
277 Testing that is_item() returns true ... Passed.
278 The cursor should be at item [59] of the sequence
279 (counting the first item as [0]). I will advance the cursor
280 to the end of the sequence, checking that each item is correct...Passed.
281 I'll call start() and look at the items one more time...
282 The cursor should be at item [0] of the sequence
283 (counting the first item as [0]). I will advance the cursor
284 to the end of the sequence, checking that each item is correct...Passed.
285 All tests passed for this sequence.
286
287 Assignment operator test: for a sequence with cursor near middle.
288 Testing that size() returns 60 ... Passed.
289 Testing that is_item() returns true ... Passed.
290 The cursor should be at item [30] of the sequence
291 (counting the first item as [0]). I will advance the cursor
292 to the end of the sequence, checking that each item is correct...Passed.
293 I'll call start() and look at the items one more time...
294 The cursor should be at item [0] of the sequence
295 (counting the first item as [0]). I will advance the cursor
296 to the end of the sequence, checking that each item is correct...Passed.
297 All tests passed for this sequence.
298
299 Assignment operator test: for a sequence with cursor near middle.
300 Testing that size() returns 60 ... Passed.
301 Testing that is_item() returns true ... Passed.
302 The cursor should be at item [0] of the sequence
303 (counting the first item as [0]). I will advance the cursor
304 to the end of the sequence, checking that each item is correct...Passed.
305 I'll call start() and look at the items one more time...
306 The cursor should be at item [0] of the sequence
307 (counting the first item as [0]). I will advance the cursor
308 to the end of the sequence, checking that each item is correct...Passed.
309 All tests passed for this sequence.
310
311 Assignment operator test: for a sequence with no current item.
312 Testing that size() returns 60 ... Passed.
313 Testing that is_item() returns false ... Passed.
314 I'll call start() and look at the items one more time...
315 All tests passed for this sequence.
316
317 Checking correctness of a self-assignment x = x;
318 Testing that size() returns 60 ... Passed.
319 Testing that is_item() returns true ... Passed.
320 The cursor should be at item [1] of the sequence
321 (counting the first item as [0]). I will advance the cursor
322 to the end of the sequence, checking that each item is correct...Passed.
323 I'll call start() and look at the items one more time...
324 The cursor should be at item [0] of the sequence
325 (counting the first item as [0]). I will advance the cursor
326 to the end of the sequence, checking that each item is correct...Passed.
327 All tests passed for this sequence.
328
329 All tests of this sixth function have been passed.
330 Test 6 got 2 points out of a possible 2.
331 END OF TEST 6.
332
333
334 START OF TEST 7:
335 Testing insert/attach when current DEFAULT_CAPACITY exceeded (3 points).
336 Testing to see that attach works correctly when the
337 current capacity is exceeded.
338 Testing that size() returns 60 ... Passed.
339 Testing that is_item() returns true ... Passed.
340 The cursor should be at item [59] of the sequence
341 (counting the first item as [0]). I will advance the cursor
342 to the end of the sequence, checking that each item is correct...Passed.
343 I'll call start() and look at the items one more time...
344 The cursor should be at item [0] of the sequence
345 (counting the first item as [0]). I will advance the cursor
```

```
346 to the end of the sequence, checking that each item is correct...Passed.
347 All tests passed for this sequence.
348
349 Testing to see that insert works correctly when the
350 current capacity is exceeded.
351 Testing that size() returns 60 ... Passed.
352 Testing that is_item() returns true ... Passed.
353 The cursor should be at item [0] of the sequence
354 (counting the first item as [0]). I will advance the cursor
355 to the end of the sequence, checking that each item is correct...Passed.
356 I'll call start() and look at the items one more time...
357 The cursor should be at item [0] of the sequence
358 (counting the first item as [0]). I will advance the cursor
359 to the end of the sequence, checking that each item is correct...Passed.
360 All tests passed for this sequence.
361
362 All tests of this seventh function have been passed.
363 Test 7 got 3 points out of a possible 3.
364 END OF TEST 7.
365
366 Your sequence implementation has scored
367 21 points out of the 21 points based on this test program.
368
```