

Source File: ~/2336/34/lab34.cpp
Input: under control of main function
Output: under control of main function
Value: 2

Add the following member functions to the `miniVector` class template. The prototypes of these functions are shown in Figure 1, a sample `main` function for testing your implementation is shown in Figure 2, and a sample execution sequence is shown in Figure 3. You will need to add a target of `lab34main` to the definition of `targets1srcfile` in your Makefile.

```

1 void insert(int i, const T& item);
2 // insert item at index i in the vector.
3 // Precondition vector is not empty and 0 <= i <= vSize.
4 // Postcondition the vector size increases by 1.
5
6 void erase(int i);
7 // erase the item at index i in the vector.
8 // Precondition vector is not empty and 0 <= i < vSize.
9 // Postcondition the vector size decreases by 1.
```

Figure 1. New Member Functions of `miniVector`

Note that the implementation of `insert()` allows `i == vSize`. The operation in this situation is equivalent to `push_back(item)`. A reallocation is necessary if `vSize == vCapacity`. Also note that for `miniVector v`, `v.erase(vSize() - 1)` is equivalent to `v.pop_back()`.

```

1 #include <iostream>
2 #include <cstdlib>
3 #include <d_vector.h>
4 #include <lab34.cpp>
5
6 using namespace std;
7
8 template <typename T>
9 ostream& operator<<(ostream& os, const miniVector<T>& v)
10 {
11     int i = 0;
12
13     if (!v.empty())
14     {
15         os << "miniVector" << endl << '{' << endl;
16         for (i = 0; i < v.size(); ++i)
17             os << "  [" << i << "] = " << v[i] << endl;
18         os << '}' << endl;
19     }
20
21     return os;
22 }
23
```

Figure 2. `/usr/local/2336/src/lab34main.C` (Part 1 of 4)

```
24 int main()
25 {
26     int i, n;
27     miniVector<int> v;
28
29     cout << "v.size() = " << v.size() << endl;
30     cout << "v.capacity() = " << v.capacity() << endl;
31
32     // On an empty vector, check to see if erase throws
33     // an underflowError exception
34     try
35     {
36         v.erase(0);
37     }
38     catch (underflowError error)
39     {
40         cerr << "**1**" << error.what() << endl;
41     }
42
43     // On an empty vector, check to see if insert throws
44     // an underflowError exception when pos is non-zero
45     try
46     {
47         v.insert(1, 2002);
48     }
49     catch (underflowError error)
50     {
51         cerr << "**2**" << error.what() << endl;
52     }
53
54     // Insert into an empty vector at position 0
55     try
56     {
57         v.insert(0, 2002);
58     }
59     catch (underflowError error)
60     {
61         cerr << "**3**" << error.what() << endl;
62     }
63     catch (indexRangeError error)
64     {
65         cerr << "**4**" << error.what() << endl;
66     }
67     cout << v << endl;
68
```

Figure 2. /usr/local/2336/src/lab34main.C (Part 2 of 4)

```
69 // Attempt an insert at an invalid position
70 try
71 {
72     v.insert(v.size() + 1, 2003);
73 }
74 catch (indexRangeError error)
75 {
76     cerr << "**5**" << error.what() << endl;
77 }
78
79 for (i = 0; i < 5; ++i)
80     v.push_back(2305);
81 cout << v << endl;
82
83 try
84 {
85     n = v.size();
86     for (i = 0; i < 2*n; i += 2)
87         v.insert(i, i);
88 }
89 catch (indexRangeError error)
90 {
91     cerr << "**6**" << error.what() << endl;
92 }
93 cout << v << endl;
94
95 // On a non-empty vector, check to see if erase throws
96 // an indexRangeError exception
97 try
98 {
99     v.erase(v.size());
100 }
101 catch (indexRangeError error)
102 {
103     cerr << "**7**" << error.what() << endl;
104 }
105
106 // On a non-empty vector, check to see if insert throws
107 // an indexRangeError exception
108 try
109 {
110     v.insert(v.size() + 1, 2002);
111 }
112 catch (indexRangeError error)
113 {
114     cerr << "**8**" << error.what() << endl;
115 }
116
```

Figure 2. /usr/local/2336/src/lab34main.C (Part 3 of 4)

```

117     try
118     {
119         while (v.size() > 0)
120         {
121             v.erase(0);
122             cout << v << endl;
123         }
124     }
125     catch (indexRangeError error)
126     {
127         cerr << "***9**" << error.what() << endl;
128     }
129
130     return EXIT_SUCCESS;
131 }

```

Figure 2. /usr/local/2336/src/lab34main.C (Part 4 of 4)

```

1  newuser@csunix ~> cd 2336
2  newuser@csunix ~/2336> ./getlab.ksh 34
3      * Checking to see if a folder exists for Lab 34. . .No
4      * Creating a folder for Lab 34
5      * Checking to see if Lab 34 has sample input and output files. . .Yes
6      * Copying input and output files for Lab 34
7          from folder /usr/local/2336/data/34 to folder ./34
8      * Checking to see if /usr/local/2336/src/lab34main.C exists. . .Yes
9      * Copying file /usr/local/2336/src/lab34main.C to folder ./34
10     * Checking to see if /usr/local/2336/include/lab34.h exists. . .No
11     * Copying file /usr/local/2336/src/Makefile to folder ./34
12     * Adding a target of lab34main to targets1srcfile
13     * Touching file ./34/lab34.cpp
14     * Edit file ./34/lab34.cpp in Notepad++
15  newuser@csunix ~/2336> cd 34
16  newuser@csunix ~/2336/34> ls
17  01.out      Makefile      lab34.cpp      lab34main.C
18  newuser@csunix ~/2336/34> make lab34main
19  g++ -g -Wall -std=c++11 -c lab34main.C -I/usr/local/2336/include -I.
20  g++ -o lab34main lab34main.o -L/usr/local/2336/lib -lm -lbits
21  newuser@csunix ~/2336/34> ./lab34main
22  v.size() = 0
23  v.capacity() = 0
24  **1**miniVector erase(): vector empty
25  **2**miniVector insert(): vector empty
26  miniVector
27  {
28      [0] = 2002
29  }
30

```

Figure 3. Commands to Compile, Link, & Run Lab 34 (Part 1 of 4)

```
31 **5**miniVector insert(): index range error index 2 size = 1
32 miniVector
33 {
34     [0] = 2002
35     [1] = 2305
36     [2] = 2305
37     [3] = 2305
38     [4] = 2305
39     [5] = 2305
40 }
41
42 miniVector
43 {
44     [0] = 0
45     [1] = 2002
46     [2] = 2
47     [3] = 2305
48     [4] = 4
49     [5] = 2305
50     [6] = 6
51     [7] = 2305
52     [8] = 8
53     [9] = 2305
54     [10] = 10
55     [11] = 2305
56 }
57
58 **7**miniVector erase(): index range error index 12 size = 12
59 **8**miniVector insert(): index range error index 13 size = 12
60 miniVector
61 {
62     [0] = 2002
63     [1] = 2
64     [2] = 2305
65     [3] = 4
66     [4] = 2305
67     [5] = 6
68     [6] = 2305
69     [7] = 8
70     [8] = 2305
71     [9] = 10
72     [10] = 2305
73 }
74
75 miniVector
76 {
77     [0] = 2
78     [1] = 2305
79     [2] = 4
80     [3] = 2305
81     [4] = 6
82     [5] = 2305
83     [6] = 8
84     [7] = 2305
```

Figure 3. Commands to Compile, Link, & Run Lab 34 (Part 2 of 4)

```
85     [8] = 10
86     [9] = 2305
87 }
88
89 miniVector
90 {
91     [0] = 2305
92     [1] = 4
93     [2] = 2305
94     [3] = 6
95     [4] = 2305
96     [5] = 8
97     [6] = 2305
98     [7] = 10
99     [8] = 2305
100 }
101
102 miniVector
103 {
104     [0] = 4
105     [1] = 2305
106     [2] = 6
107     [3] = 2305
108     [4] = 8
109     [5] = 2305
110     [6] = 10
111     [7] = 2305
112 }
113
114 miniVector
115 {
116     [0] = 2305
117     [1] = 6
118     [2] = 2305
119     [3] = 8
120     [4] = 2305
121     [5] = 10
122     [6] = 2305
123 }
124
125 miniVector
126 {
127     [0] = 6
128     [1] = 2305
129     [2] = 8
130     [3] = 2305
131     [4] = 10
132     [5] = 2305
133 }
134
135 miniVector
136 {
137     [0] = 2305
138     [1] = 8
```

Figure 3. Commands to Compile, Link, & Run Lab 34 (Part 3 of 4)

```
139     [2] = 2305
140     [3] = 10
141     [4] = 2305
142 }
143
144 miniVector
145 {
146     [0] = 8
147     [1] = 2305
148     [2] = 10
149     [3] = 2305
150 }
151
152 miniVector
153 {
154     [0] = 2305
155     [1] = 10
156     [2] = 2305
157 }
158
159 miniVector
160 {
161     [0] = 10
162     [1] = 2305
163 }
164
165 miniVector
166 {
167     [0] = 2305
168 }
169
170
171 newuser@csunix ~/2336/34> ./lab34main > my.out 2>&1
172 newuser@csunix ~/2336/34> diff 01.out my.out
173 newuser@csunix ~/2336/34>
```

Figure 3. Commands to Compile, Link, & Run Lab 34 (Part 4 of 4)