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`.

```
  void insert(int i, const T& item);
  // insert item at index i in the vector.
  // Precondition vector is not empty and 0 <= i <= vSize.
  // Postcondition the vector size increases by 1.

  void erase(int i);
  // erase the item at index i in the vector.
  // Precondition vector is not empty and 0 <= i < vSize.
  // 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()`.

```
#include <iostream>
#include <cstdlib>
#include <d_vector.h>
#include <lab34.cpp>

using namespace std;

template <typename T>
ostream& operator<<(ostream& os, const miniVector<T>& v)
{
  int i = 0;

  if (!v.empty())
  {
    os << "miniVector" << endl << ’{’ << endl;
    for (i = 0; i < v.size(); ++i)
      os << "  [" << i << "] = " << v[i] << endl;
    os << ’}’ << endl;
  }

  return os;
}

```

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

```
int main()
{
  int i, n;
  miniVector<int> v;

  cout << "v.size() = " << v.size() << endl;
  cout << "v.capacity() = " << v.capacity() << endl;

  // On an empty vector, check to see if erase throws
  // an underflowError exception
  try
  {
    v.erase(0);
  }
  catch (underflowError error)
  {
    cerr << "**1**" << error.what() << endl;
  }

  // On an empty vector, check to see if insert throws
  // an underflowError exception when pos is non-zero
  try
  {
    v.insert(1, 2002);
  }
  catch (underflowError error)
  {
    cerr << "**2**" << error.what() << endl;
  }

  // Insert into an empty vector at position 0
  try
  {
    v.insert(0, 2002);
  }
  catch (underflowError error)
  {
    cerr << "**3**" << error.what() << endl;
  }
  catch (indexRangeError error)
  {
    cerr << "**4**" << error.what() << endl;
  }
  cout << v << endl;

```

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

```
  // Attempt an insert at an invalid position
  try
  {
    v.insert(v.size() + 1, 2003);
  }
  catch (indexRangeError error)
  {
    cerr << "**5**" << error.what() << endl;
  }

  for (i = 0; i < 5; ++i)
    v.push_back(2305);
  cout << v << endl;

  try
  {
    n = v.size();
    for (i = 0; i < 2*n; i += 2)
      v.insert(i, i);
  }
  catch (indexRangeError error)
  {
    cerr << "**6**" << error.what() << endl;
  }
  cout << v << endl;

  // On a non-empty vector, check to see if erase throws
  // an indexRangeError exception
  try
  {
    v.erase(v.size());
  }
  catch (indexRangeError error)
  {
    cerr << "**7**" << error.what() << endl;
  }

  // On a non-empty vector, check to see if insert throws
  // an indexRangeError exception
  try
  {
    v.insert(v.size() + 1, 2002);
  }
  catch (indexRangeError error)
  {
    cerr << "**8**" << error.what() << endl;
  }

```

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

```
  try
  {
    while (v.size() > 0)
    {
      v.erase(0);
      cout << v << endl;
    }
  }
  catch (indexRangeError error)
  {
    cerr << "**9**" << error.what() << endl;
  }

  return EXIT_SUCCESS;
}
```

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

```
newuser@csunix ~> cd 2336
newuser@csunix ~/2336> ./getlab.ksh 34
  * Checking to see if a folder exists for Lab 34. . .No
  * Creating a folder for Lab 34
  * Checking to see if Lab 34 has sample input and output files. . .Yes
  * Copying input and output files for Lab 34
    from folder /usr/local/2336/data/34 to folder ./34
  * Checking to see if /usr/local/2336/src/lab34main.C exists. . .Yes
  * Copying file /usr/local/2336/src/lab34main.C to folder ./34
  * Checking to see if /usr/local/2336/include/lab34.h exists. . .No
  * Copying file /usr/local/2336/src/Makefile to folder ./34
  * Adding a target of lab34main to targets1srcfile
  * Touching file ./34/lab34.cpp
  * Edit file ./34/lab34.cpp in Notepad++
newuser@csunix ~/2336> cd 34
newuser@csunix ~/2336/34> ls
01.out       Makefile     lab34.cpp    lab34main.C
newuser@csunix ~/2336/34> make lab34main
g++ -g -Wall -std=c++11 -c lab34main.C -I/usr/local/2336/include -I.
g++ -o lab34main lab34main.o -L/usr/local/2336/lib -lm -lbits
newuser@csunix ~/2336/34> ./lab34main
v.size() = 0
v.capacity() = 0
**1**miniVector erase(): vector empty
**2**miniVector insert(): vector empty
miniVector
{
  [0] = 2002
}

```

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

```
**5**miniVector insert(): index range error  index 2  size = 1
miniVector
{
  [0] = 2002
  [1] = 2305
  [2] = 2305
  [3] = 2305
  [4] = 2305
  [5] = 2305
}

miniVector
{
  [0] = 0
  [1] = 2002
  [2] = 2
  [3] = 2305
  [4] = 4
  [5] = 2305
  [6] = 6
  [7] = 2305
  [8] = 8
  [9] = 2305
  [10] = 10
  [11] = 2305
}

**7**miniVector erase(): index range error  index 12  size = 12
**8**miniVector insert(): index range error  index 13  size = 12
miniVector
{
  [0] = 2002
  [1] = 2
  [2] = 2305
  [3] = 4
  [4] = 2305
  [5] = 6
  [6] = 2305
  [7] = 8
  [8] = 2305
  [9] = 10
  [10] = 2305
}

miniVector
{
  [0] = 2
  [1] = 2305
  [2] = 4
  [3] = 2305
  [4] = 6
  [5] = 2305
  [6] = 8
  [7] = 2305
```

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

```
  [8] = 10
  [9] = 2305
}

miniVector
{
  [0] = 2305
  [1] = 4
  [2] = 2305
  [3] = 6
  [4] = 2305
  [5] = 8
  [6] = 2305
  [7] = 10
  [8] = 2305
}

miniVector
{
  [0] = 4
  [1] = 2305
  [2] = 6
  [3] = 2305
  [4] = 8
  [5] = 2305
  [6] = 10
  [7] = 2305
}

miniVector
{
  [0] = 2305
  [1] = 6
  [2] = 2305
  [3] = 8
  [4] = 2305
  [5] = 10
  [6] = 2305
}

miniVector
{
  [0] = 6
  [1] = 2305
  [2] = 8
  [3] = 2305
  [4] = 10
  [5] = 2305
}

miniVector
{
  [0] = 2305
  [1] = 8
```

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

```
  [2] = 2305
  [3] = 10
  [4] = 2305
}

miniVector
{
  [0] = 8
  [1] = 2305
  [2] = 10
  [3] = 2305
}

miniVector
{
  [0] = 2305
  [1] = 10
  [2] = 2305
}

miniVector
{
  [0] = 10
  [1] = 2305
}

miniVector
{
  [0] = 2305
}


newuser@csunix ~/2336/34> ./lab34main > my.out 2>&1
newuser@csunix ~/2336/34> diff 01.out my.out
newuser@csunix ~/2336/34>
```

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