Self-Evaluation Report for Programming Assignments

Section I: Basic Information

1. **Programming assignment #: 5A**
2. **Name of the author: Esther Choi**
3. **Name of the peer reviewer, if any: Priscilla Son**
4. **Due date of the assignment: April 5, 2017**
5. **Date when the assignment was finished: April 3, 2017**
6. **Number of hours spent in programming**: **roughly 5 hours**

Section II: Integrity Review

Integrity rules for regular programming assignments

- **Peer discussion**: Peer discussion of code shown on a screen or board is acceptable for explanation of ideas and for debugging purpose. Such discussion may help to cultivate an open learning environment in the class, but you should carefully read the guidelines below to avoid any dishonest behavior and never step over the guidelines explicitly described in the following.
- **Never use any code (i.e. C++ statements, segments of a program or an entire program) written by others (except for examples in our textbooks or reading)**: Any copy-and-paste of code from other people's programs or from websites is viewed as cheating and you will get 0 points for the assignment.
- **Never circulate your code to others**: You should never pass around your code (electronically or on paper) to others except for the TA and the instructor. Violating this rule is viewed as cheating in the class and the provider will receive 0 points for the assignment.
- **Never provide false or exaggerated results of test cases**: You need to report results of test cases in the self-evaluation report together with all your source code files for each assignment**.** Providing false or exaggerated results of test cases in the report is viewed as cheating and you will receive 0 points for the assignment.
- **Demonstrate the credibility of your authorship of the work**: When you submit your code as your own work for points, you should make sure that you are able to explain your code and reconstruct your code from scratch without any outside help when requested. If you are not able to do that on your own when requested, you will get 0 points for the assignment and there will be an investigation.
- **Consequence of cheating in the class**: Cheating ends in 0 points for the assignments followed by disciplinary actions as described in the student handbook.

1. **Have you ever received any code written by others? No**
2. **Have you ever passed any code you wrote to others? No**
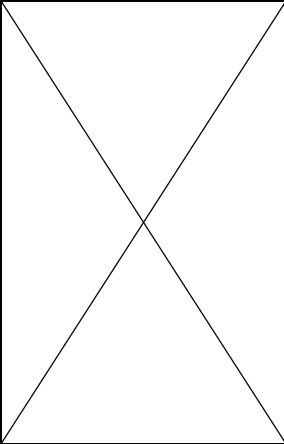3. **Have you ever used any code written by others? No**
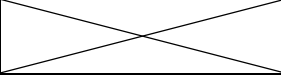
Section III: Test cases and peer review

Note: To get all the points, you should have a peer reviewer watch the behavior of your program before you submit the work. You should prepare your own test cases and have your reviewer see the results when you run your program over the test cases. Optionally, you may also have the reviewer run your program through the reviewer's own test cases to see whether your program works correctly.

1. **Compile and run your code using Visual C++ 2013 as the testing environment and describe the test cases used and the results you and the reviewer have observed**:

K		
Input (month, day, year)	**Testing For**	**Result(s) (after pressing 'D' to display)**
2, 29, 2000	Pushing back correct date	Message displays, indicating that date was successfully pushed into the vector Vector contains the following 1 dates: 2/29/2000
2, 29, 1999	Attempt to push back an incorrect date	Error message displays and program loops until correct date entry is entered
0, 0, 0	Attempt to push back another incorrect date	Error message displays and program loops until correct date entry is entered

1, 11, 1956	Pushing back another correct date	Message displays, indicating that date was successfully pushed into the vector
		
		Vector contains the following 2 dates:
		  2/29/2000
		  1/11/1956

B		
Input (filename)	**Testing For**	**Result(s) (after pressing 'D' to display)**
t.txt	An incorrect and/or nonexistent filename	Error message displays indicating an incorrect filename and program loops until correct filename is entered
dates.txt	A correct filename with a positive number of records	Message displays indicating that the dates in the file have successfully been pushed into the vector Vector contains the following 6 dates: 1/14/2000

		2/12/1990
		1/1/2000
		3/2/1987
		6/30/1990
		1/1/2000
		Note how the dates pushed in through 'K' are no longer present in the vector

K (Redo)		
Input (month, day, year)	**Testing For**	**Result(s) (after pressing 'D' to display)**
8, 30, 2008	Pushing back a correct date	Vector contains the following 7 dates: 1/14/2000 2/12/1990 1/1/2000 3/2/1987 6/30/1990 1/1/2000 8/30/2008

M		
Input (i) (month, day, year)	**Testing For**	**Result(s) (after pressing 'D' to display)**
3 4, 5, 1845	Modifying date element at 3rd index	Vector contains the following 7 dates: 1/14/2000 2/12/1990 1/1/2000 4/5/1845 6/30/1990 1/1/2000 8/30/2008
-1	Negative index entry	Error message displays indicating an invalid index; user may enter menu value 'M' and retry
10	Attempting to modify element at a nonexistent index	Error message displays indicating an invalid index; user may enter menu value 'M' and retry

F		
Input (dBegin, dEnd)	**Testing For**	**Result(s)**
1, 1, 1990 1, 1, 2000	Whether 'F' is working as it should and	The following dates fall within the specified range: 2/12/1990

Input	Testing For	Result(s)
	displaying only the proper results	1/1/2000 6/30/1990 1/1/2000
1, 1, 1 2, 2, 2	What displays when there are no dates between the given ranges	The following dates fall within the specified range: None :(

S		

Input	Testing For	Result(s) (after pressing 'D' to display)

	Whether the vector can sort successfully	Message displays indicating successful completion of the sort; user may enter 'D' to view the results of the mergeSort()
		
		Vector contains the following 7 dates:
		  4/5/1845
		  2/12/1990
		  6/30/1990
		  1/1/2000
		  1/1/2000
		  1/14/2000
		  8/30/2008

F		

Input	Testing For	Result(s)

1, 1, 1990
1, 1, 2000
	Whether 'F' works properly on a sorted vector	The following dates fall within the specified range:
		  2/12/1990
		  6/30/1990
		  1/1/2000
		  1/1/2000

W		

Input (output filename)	Testing For	Result(s) (in output file)
test.txt	Whether the vector contents are successfully written to output file	Message displays indicating successful writing of vector elements to output file; user may open output file to view contents 7 4 5 1845 2 12 1990 6 30 1990 1 1 2000

		1 1 2000 1 14 2000 8 30 2008
D		
Input	**Testing For**	**Result(s)**
✕	Displaying when a vector is empty	Vector contains the following 0 dates: Nothing :(
B (Redo)		
Input (filename, numDateRecords)	**Testing For**	**Result(s) (after pressing 'D' to display)**
dates.txt, -6	When numDateRecords is a negative value	Error message displays indicating that the operation was unsuccessful and that numDateRecords must be a non-negative value Vector contains the following dates: Nothing :(
dates.txt, 0	When numDateRecords is 0	Vector contains the following 0 dates: Nothing :(

2. **Description of bugs or other problems discovered by you or the peer reviewer, if any:**

   No bugs or other problems were discovered by us.

3. **Have you implemented everything required by the programming assignment? If not, describe what are missing.**

   Yes, I have implemented everything that was required of us for #5A, according to the specifications in the PDF file.

Section IV Self-evaluation: Points you think you deserve <u>6</u>

- **Deduct one point if you submit the work after the due date but before it's closed.**

- **Grading scale:**

0. Nothing done **or missing the self-evaluation report or missing the integrity review** in the report

1. Source code is completed but the code fails to compile successfully
2. Source code can compile and do something required, but has serious bugs or miss a couple of key features.
3. Source code can compile and do most of the features required, but has many minor bugs or miss a key required feature.
4. Source code can compile and do all the features required, nearly fully functional, only a couple of minor bugs.
5. Source code can compile and do all the features required, fully functional, no bugs.
6. **In addition to the points received according to the rubrics above, get one more point** if

    a. **the self-evaluation report contains sufficient descriptions of test cases used (0.25 point)**, and

    b. **the self-evaluation report indicates the results of the test cases were verified by a peer reviewer (0.25 point)**, and

    c. **the source code is well indented and commented to make it visually very readable (0.5 point)**.