Homework #1: Testing the performance of linked lists.

Testing the performance of linked lists for managing a large number of dates:

**Step 1.** Implement additional elements as described in the following into your program for Programming #5B:

- **Option X (do random InsertInOrder for n times):**
  Add one option X into the menu. When the user selects this option, your program should (i) call the Clear method to empty the linked list of your date database, (ii) ask the user to enter a natural number \( n \), (iii) declare a local DateType object \( d \), and (iv) set up a loop to go through \( n \) iterations and in each iteration call \( d.SetRandomDate() \) to set a random date and then call the InsertInOrder method of the linked list class to insert the date in \( d \) into the linked list of your date database.

- **Option Y (do random Insert for n times):**
  Add one option Y into the menu. When the user selects this option, your program should (i) call the Clear method to empty the linked list of your date database, (ii) ask the user to enter a natural number \( n \), (iii) declare a local DateType object \( d \), and (iv) set up a loop to go through \( n \) iterations and in each iteration call \( d.SetRandomDate() \) to set a random date and then call the Insert method of the linked list class to insert the date in \( d \) into the linked list of your date database.

- **Option Z (do random Remove for m times):**
  Add one option Z into the menu. When the user selects this option, your program should (i) ask the user to enter a natural number \( m \), (ii) declare a local DateType object \( d \), and (iii) set up a loop to go through \( m \) iterations and in each iteration first call \( d.SetRandomDate() \) to set a random date and then call Remove(\( d \)) to try to remove the date in \( d \) from the linked list of your date database.

- Test to make sure your implementation of the two additional options is fine.

**Step 2. Experiments:**

**A. Test and report the time needed for n insertions into a sorted linked list:** Try option X several times and use different values of \( n \) from 1000, 10000, 100000, and up to at least 200,000 or higher. Each time use your watch to roughly estimate the amount of time option X takes (to insert \( n \) random dates into a sorted linked list). **Record and report your findings.**

**B. Right after Experiment A, test and report the time needed for m deletions in a sorted linked list of about n nodes (where n is the value you used for Option X in the very end of Experiment A):** Try option Z several times now using different values of \( m \) from 1000, 10000, 100000, and up as you did in Experiment A above. Each time use your watch to roughly estimate the amount of time option Z takes (to remove \( n \) random dates from a
sorted linked list established by Option X in the very end of Experiment A). **Record and report your findings.**

C. Test and report the time needed for \( n \) insertions in an unsorted linked list: Try option Y several times and use different values of \( n \) from 1000, 10000, 100000, and up to at least 1,000,000 or higher. Each time use your watch to roughly estimate the amount of time option Y takes (to insert \( n \) random dates into an unsorted linked list). **Record and report your findings.**

D. Right after Experiment C, Test and report the time needed for \( m \) deletions in an unsorted linked list of about \( n \) nodes (where \( n \) is the value you used for Option Y in the very end of Experiment C): Try option Z several times now using different values of \( m \) from 1000, 10000, 100000, and up as you did in Experiment C above. Each time use your watch to roughly estimate the amount of time option Z takes (to remove \( n \) random dates from an unsorted linked list established by Option Y in the very end of Experiment C). **Record and report your findings.**

**Step 3. Reflection and analysis:**

A. About the time needed for \( n \) insertions into an initially empty sorted linked list: What do you think is the relationship between the size \( n \) and the amount of time needed? Why? **Record your thoughts/analysis.**

B. About the time needed for \( m \) deletions in a sorted linked list of about \( n \) nodes: What do you think is the relationship between the size \( n \) and the size \( m \) and the amount of time needed? Why? **Record and report your findings.**

C. About the time needed for \( n \) insertions into an initially unsorted linked list: What do you think is the relationship between the size \( n \) and the amount of time needed? Why? **Record and report your findings.**

D. About the time needed for \( m \) deletions in an unsorted linked list of about \( n \) nodes: What do you think is the relationship between the size \( n \) and the size \( m \) and the amount of time needed? Why? **Record and report your findings.**

**Submit your work**

- Record all your experimental findings in Step 2 and your thoughts in Step 3 above in a WORD document. Submit the WORD document under Canvas.
- Compress your entire Program folder into a zip file and upload it through Biola Canvas.
- Carefully fill out this self-evaluation report and upload it through Biola Canvas. Note that you will receive no point for missing the self-evaluation report or missing the integrity review in the report.