A doubly linked list – Testing

Test your functionalities properly and seriously. If your implementation and timing should work properly and correctly. If your timing does not match with your code or does not work, you will not get a full credit or even get a penalty for your implementation.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Step | Operations | Points | Self  Grading | comments |
| 1-1  1-2 | push and pop commands  including  pop\_all\* | 1 |  |  |
| 2 | half() and show()\* | 1 |  |  |
| 3 | sort commands  selectionSort()  insertionSort()  sorted() | 1 |  |  |
| 4-1 | push\_sorted()\* | 1 |  |  |
| 4-2 | push\_sortedN() | 1 |  |  |
| 5 | push\_sortedNlog()\*\* | 1 |  |  |
| 6 | unique()\* | 1 |  |  |
| 7 | reverse() | 1 |  |  |
| 8 | shuffle()\*\* | 1 |  |  |
| 9 | Timing Test &  Analysis | -2 to +1 |  |  |
|  | Total | 10 |  |  |

**Test Hint 1: pop\_all()**

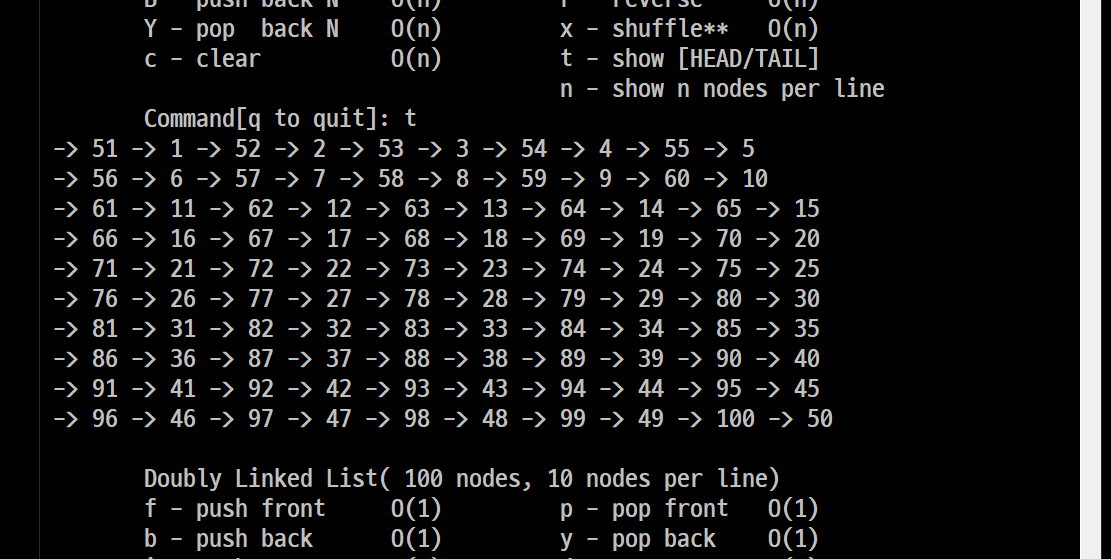
To test pop\_all(), you may need to generate a sequence that has a consecutive numbers of a certain value. You may use "push back N" command option with a negative N provided.

If you want to make a sequence of, for example, 3 2 1 4 5 ... ( ten thousands of 7)

* make 3 2 1 4 5 ... part first,
* select "push\_back\_N" command, enter "-10000" for N, enter "7" for a value.

**Test Hint 2: unique(), sort(),show(),pop(), push\_sorted(), push\_backN(), ...**

Try to make a sequence of numbers from 1 to 100 as shown below in a fewer steps possible. Then you may need to use all kinds of commands you implemented so far.



**Step 9:** **Test and Analysis.** You are responsible for present a sort of table or graph with clear description in Korean such that the grader can understand the timing analysis at once.

1. Time it for 100,000 nodes and estimate the elapsed time for 1 million node for following functions. Compare the timing with listdblx.exe.
   1. Selection sort, Insertion sort and quick sort.
      1. Unsorted list case
      2. Sorted list case
2. First, measure time to merge two lists. The first one is a sorted list of 50,000 nodes and the other is a sorted array of 50,000 random numbers. Estimate the elapsed time to merge two lists. Each list has one half million of random numbers sorted: Compare the timing with listdblx.exe.
   1. push\_sortedN()
   2. push\_sortedNlog()