// On my honor, I pledge that I have neither received nor provided improper assistance in the completion of this assignment.

// Signed: 조다빈 Student Number: 22000711

NOWIC afdffd7dbca0add0e8643ffbea667e6c

A doubly linked list – Self Testing

Test your functionalities properly and seriously.
Mark Pass or Fail for each operation. 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	Point	Testing	comments
1	find, more, less	0.5	PASS	Check the code. find() use one while loop, but not if Use push command to test find().
2	push commands push()	0.5	PASS	Check the code. push() must use find() and insert(), not more than 3~4 lines of code Don't add a new node if the position x is not found
3	pop commands pop_all*	0.5	PASS	Test it with over 100,000 samples. Make sure O(n), not O(n^2) Use testing method described below
4	half() and show()*	0.5	PASS	Check the correctness the middle node. half() is used in shuffle() and show() Test it with both odd & even number sequences. Record the timing of half() for 20 million samples displayed using show HEAD/TAIL on exiting. Method 1: 0.066205 sec Method 2: 0.048796 sec Method 2 is faster than Method 1 by ____26.3_____ %
5	**swap_pairs**	**0.0**	**PASS**	Check the code. It must go through the list once, not twice nor more. Test it with both odd & even number sequences.
6	sorted()	0.5	PASS	It is checked by other operations.
7	push_sorted()	0.5	PASS	Check it with unsorted, ascending and descending ordered lists. Make sure duplicated ones included such as 3 5 5 7 9 9 9. Use "reverse" menu option. Test it with over 100,000 samples. pushN to generate samples and push_sorted 100001. Make sure O(n), not O(n^2) Additionally, use testing method described below
8	unique()*	0.5	PASS	Test it with over 100,000 samples. Make sure O(n), not O(n^2) Use testing method described below
9	**reverse()**	**0.0**	**PASS**	Test it with over 100,000 samples. Make sure O(n), not O(n^2)
10	**randomize()**	**0.0**	**PASS**	Test it with over 100,000 samples. Make sure O(n), not O(n^2) The commands sort & quicksort uses randomize().

11	shuffle()*	0.5	PASS	Check the exactness. Test it with both odd & even number sequences.
	Total	4.0	0.8 + 0.2	Extra 0.1 p per step for a proper testing Extra 0.2 if you get them all right

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.

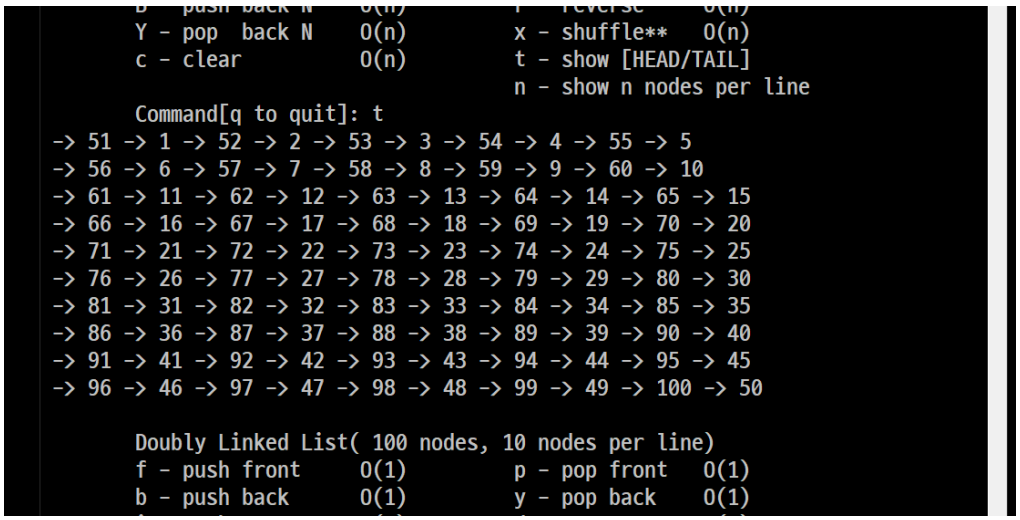
For example, make a sequence with ten thousands and another ten thousands of 7 samples:

- select "push_back_N" and enter 10,000 for random samples
- select "push_back_N" and enter -10000, then enter "7" for a value.
- run "pop-all" 7.

Test Hint 2: unique()
To test unique(), you may also need to generate a sorted sequence with consecutive numbers of a certain value. You may use "push back N" command option with a negative N provided.

For example, make a sequence with thirty thousands for each 1, 5, and 7, and ten thousands for 9, respectively and run unique().

Test Hint 3: show(), pop(), push_sorted(), push_backN(), ...
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.

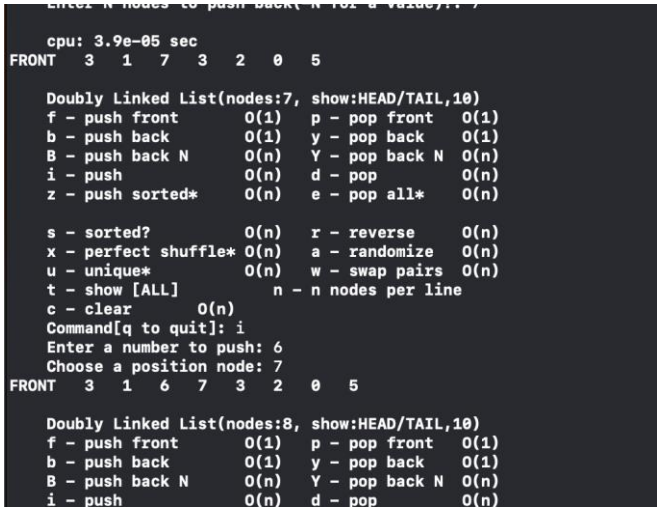


// On my honor, I pledge that I have neither received nor provided improper assistance in the completion of this assignment.

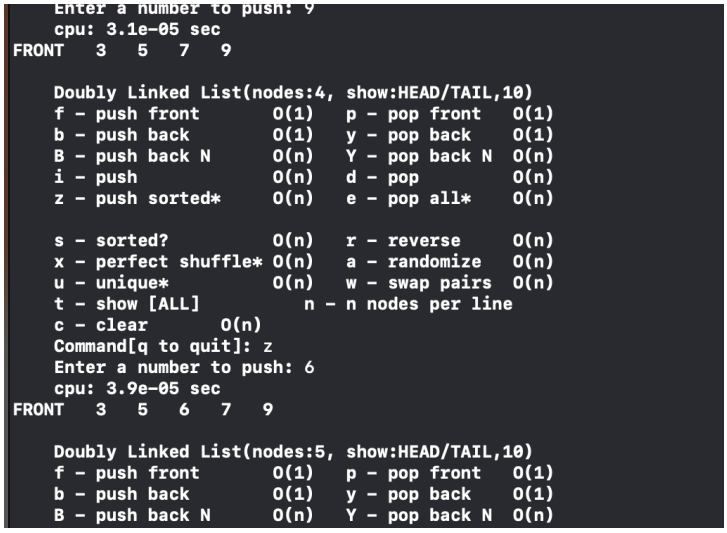
// Signed: 조다빈 Student Number: 22000711

Step1. Find, more, less

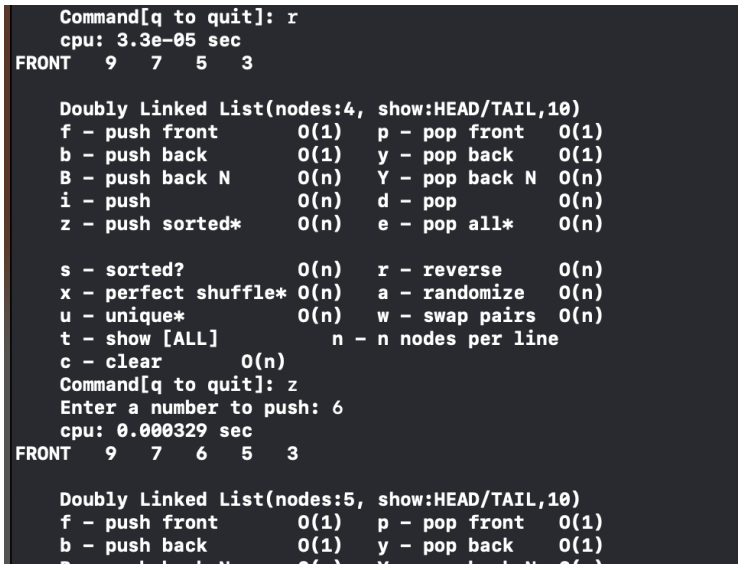
Find :



More :



Less :

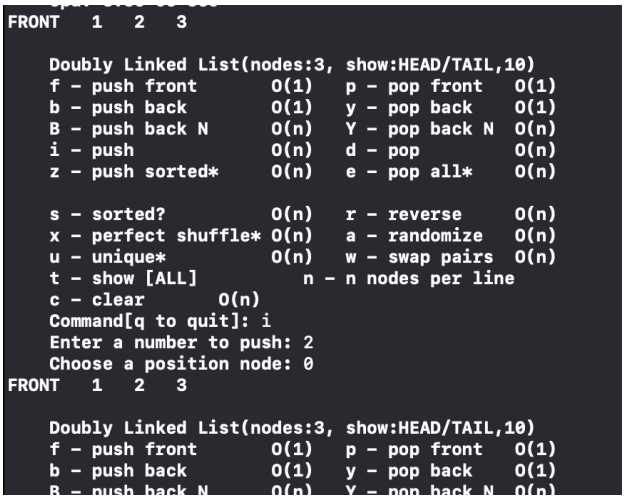


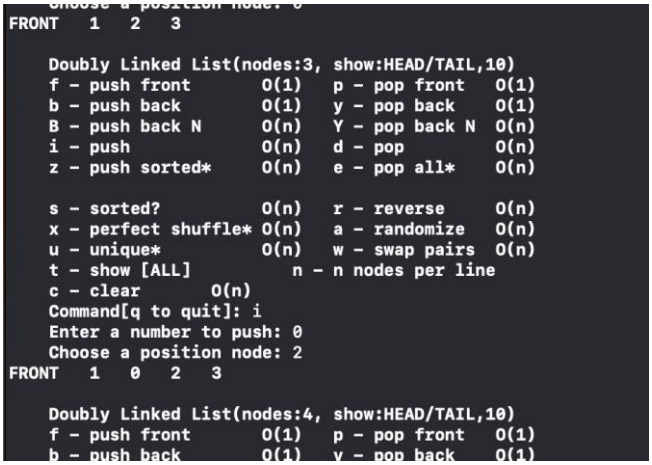
// On my honor, I pledge that I have neither received nor provided improper assistance in the completion of this assignment.

// Signed: 조다빈 Student Number: 22000711

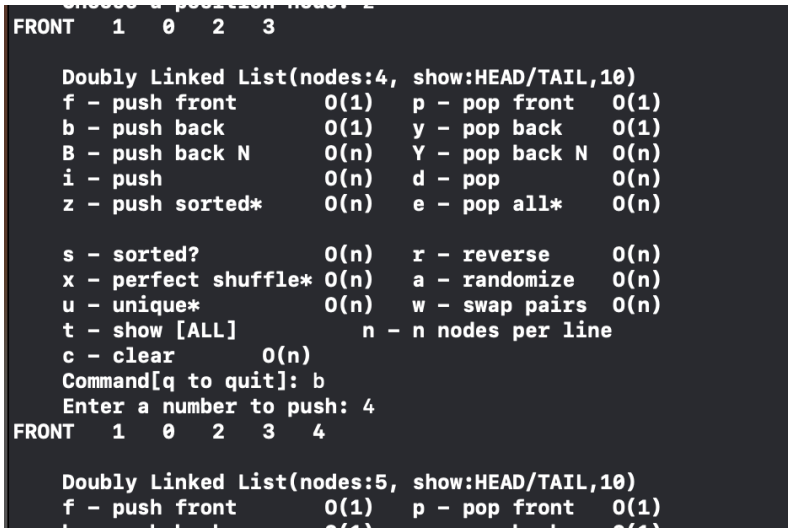
Step2. Push commands

Push() :

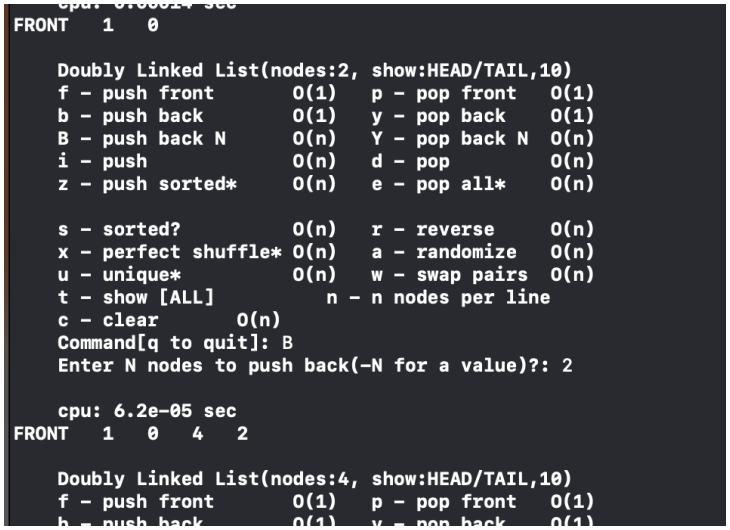




Push_back() :



Push_back(int N) :



Push_back(int N, int value) :

```
FRONT   1   0   4   2

    Doubly Linked List(nodes:4, show:HEAD/TAIL,10)
    f - push front       O(1)   p - pop front   O(1)
    b - push back        O(1)   y - pop back    O(1)
    B - push back N      O(n)   Y - pop back N  O(n)
    i - push             O(n)   d - pop         O(n)
    z - push sorted*     O(n)   e - pop all*    O(n)

    s - sorted?          O(n)   r - reverse     O(n)
    x - perfect shuffle* O(n)   a - randomize   O(n)
    u - unique*          O(n)   w - swap pairs  O(n)
    t - show [ALL]          n - n nodes per line
    c - clear       O(n)
    Command[q to quit]: B
    Enter N nodes to push back(-N for a value)?: -3
    Enter a value to push back?: 3

    pushing [4]=3
    cpu: 0.000112 sec
FRONT   1   0   4   2   3   3   3

    Doubly Linked List(nodes:7, show:HEAD/TAIL,10)
```

Push_front() :

```
    cpu: 2.4e-05 sec
FRONT   3   3   3

    Doubly Linked List(nodes:3, show:HEAD/TAIL,10)
    f - push front       O(1)   p - pop front   O(1)
    b - push back        O(1)   y - pop back    O(1)
    B - push back N      O(n)   Y - pop back N  O(n)
    i - push             O(n)   d - pop         O(n)
    z - push sorted*     O(n)   e - pop all*    O(n)

    s - sorted?          O(n)   r - reverse     O(n)
    x - perfect shuffle* O(n)   a - randomize   O(n)
    u - unique*          O(n)   w - swap pairs  O(n)
    t - show [ALL]          n - n nodes per line
    c - clear       O(n)
    Command[q to quit]: f
    Enter a number to push: 1
FRONT   1   3   3   3

    Doubly Linked List(nodes:4, show:HEAD/TAIL,10)
    f - push front       O(1)   p - pop front   O(1)
```

Step3. Pop commands

Pop() :

```
FRONT   1   2   3

    Doubly Linked List(nodes:3, show:HEAD/TAIL,10)
    f - push front       O(1)   p - pop front   O(1)
    b - push back        O(1)   y - pop back    O(1)
    B - push back N      O(n)   Y - pop back N  O(n)
    i - push             O(n)   d - pop         O(n)
    z - push sorted*     O(n)   e - pop all*    O(n)

    s - sorted?          O(n)   r - reverse     O(n)
    x - perfect shuffle* O(n)   a - randomize   O(n)
    u - unique*          O(n)   w - swap pairs  O(n)
    t - show [ALL]          n - n nodes per line
    c - clear       O(n)
    Command[q to quit]: d
    Enter a number to pop: 2
    cpu: 1.8e-05 sec
FRONT   1   3

    Doubly Linked List(nodes:2, show:HEAD/TAIL,10)
    f - push front       O(1)   p - pop front   O(1)
```

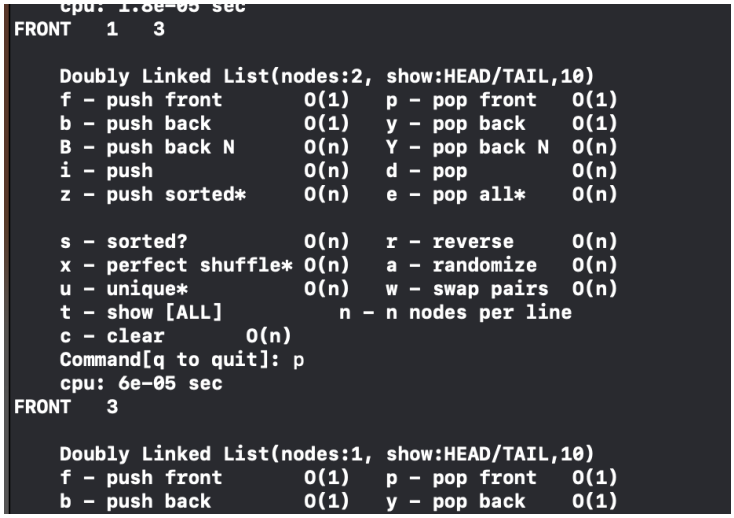
```
    cpu: 3.8e-05 sec
FRONT   1   4   1   3

    Doubly Linked List(nodes:4, show:HEAD/TAIL,10)
    f - push front       O(1)   p - pop front   O(1)
    b - push back        O(1)   y - pop back    O(1)
    B - push back N      O(n)   Y - pop back N  O(n)
    i - push             O(n)   d - pop         O(n)
    z - push sorted*     O(n)   e - pop all*    O(n)

    s - sorted?          O(n)   r - reverse     O(n)
    x - perfect shuffle* O(n)   a - randomize   O(n)
    u - unique*          O(n)   w - swap pairs  O(n)
    t - show [ALL]          n - n nodes per line
    c - clear       O(n)
    Command[q to quit]: d
    Enter a number to pop: 2
    cpu: 2.1e-05 sec
FRONT   1   4   1   3

    Doubly Linked List(nodes:4, show:HEAD/TAIL,10)
```

// On my honor, I pledge that I have neither received nor provided improper assistance in the completion of this assignment.

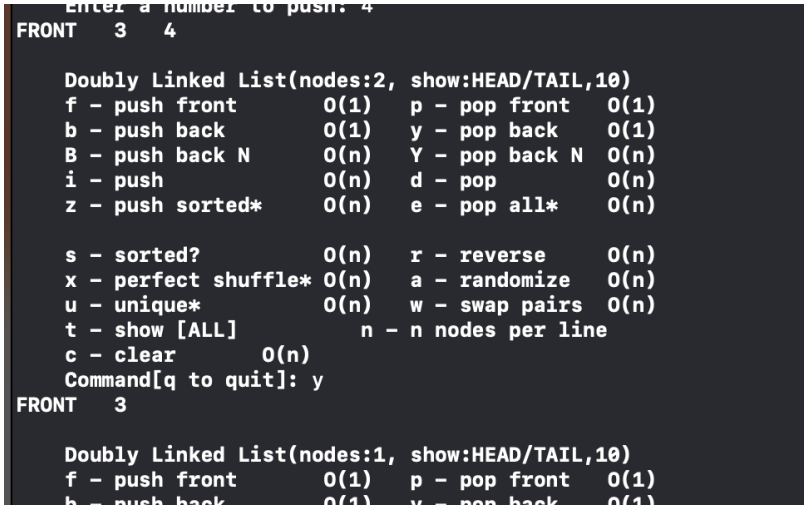
// Signed: 조다빈 Student Number: 22000711

NOWIC afdffd7dbca0add0e8643ffbea667e6c

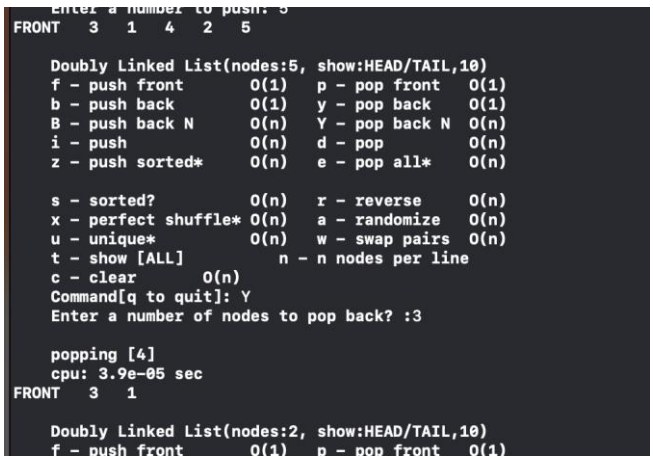
Pop_front() :

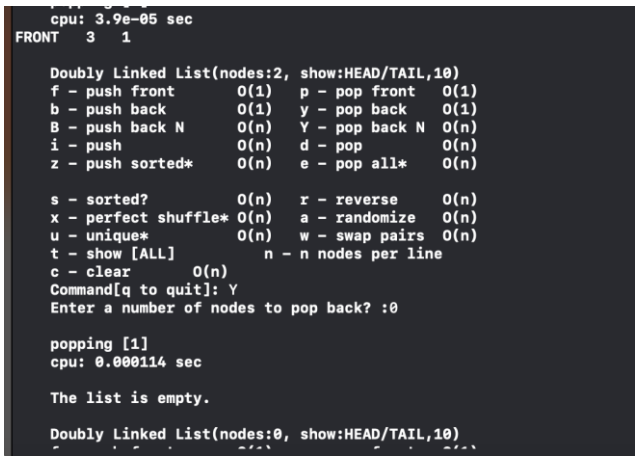


Pop_back() :



Pop_backN() :





// On my honor, I pledge that I have neither received nor provided improper assistance in the completion of this assignment.

// Signed: 조다빈 Student Number: 22000711

NOWIC afdffd7dbca0add0e8643ffbea667e6c

```
    cpu: 4e-05 sec
FRONT   1   2   0   0

    Doubly Linked List(nodes:4, show:HEAD/TAIL,10)
    f - push front       O(1)   p - pop front   O(1)
    b - push back        O(1)   y - pop back    O(1)
    B - push back N      O(n)   Y - pop back N  O(n)
    i - push             O(n)   d - pop         O(n)
    z - push sorted*     O(n)   e - pop all*    O(n)

    s - sorted?          O(n)   r - reverse     O(n)
    x - perfect shuffle* O(n)   a - randomize   O(n)
    u - unique*          O(n)   w - swap pairs  O(n)
    t - show [ALL]          n - n nodes per line
    c - clear       O(n)
    Command[q to quit]: Y
    Enter a number of nodes to pop back? :10

    popping [3]
    cpu: 4.6e-05 sec

    The list is empty.

    Doubly Linked List(nodes:0, show:HEAD/TAIL,10)
    f - push front       O(1)   p - pop front   O(1)
```

Pop_all() :

```
    The list is empty.

    Doubly Linked List(nodes:0, show:HEAD/TAIL,10)
    f - push front       O(1)   p - pop front   O(1)
    b - push back        O(1)   y - pop back    O(1)
    B - push back N      O(n)   Y - pop back N  O(n)
    i - push             O(n)   d - pop         O(n)
    z - push sorted*     O(n)   e - pop all*    O(n)

    s - sorted?          O(n)   r - reverse     O(n)
    x - perfect shuffle* O(n)   a - randomize   O(n)
    u - unique*          O(n)   w - swap pairs  O(n)
    t - show [ALL]          n - n nodes per line
    c - clear       O(n)
    Command[q to quit]: B
    Enter N nodes to push back(-N for a value)?: 50000

    cpu: 0.005411 sec
FRONT   44338   28912   27160   44027   48676   8837    46601   48091   37737   16823
    ...  5560  ...
    24854   33993   28952   27273   13348   44403   5564    14628   8604    5052

    Doubly Linked List(nodes:50000, show:HEAD/TAIL,10)
    f - push front       O(1)   p - pop front   O(1)
    b - push back        O(1)   y - pop back    O(1)
```

```
    cpu: 0.005411 sec
FRONT   44338   28912   27160   44027   48676   8837    46601   48091   37737   16823
    ...  5560  ...
    24854   33993   28952   27273   13348   44403   5564    14628   8604    5052

    Doubly Linked List(nodes:50000, show:HEAD/TAIL,10)
    f - push front       O(1)   p - pop front   O(1)
    b - push back        O(1)   y - pop back    O(1)
    B - push back N      O(n)   Y - pop back N  O(n)
    i - push             O(n)   d - pop         O(n)
    z - push sorted*     O(n)   e - pop all*    O(n)

    s - sorted?          O(n)   r - reverse     O(n)
    x - perfect shuffle* O(n)   a - randomize   O(n)
    u - unique*          O(n)   w - swap pairs  O(n)
    t - show [ALL]          n - n nodes per line
    c - clear       O(n)
    Command[q to quit]: B
    Enter N nodes to push back(-N for a value)?: -50000
    Enter a value to push back?: 7

    pushing [50000]=7
    pushing [60000]=7
    pushing [70000]=7
    pushing [80000]=7
    pushing [90000]=7
    cpu: 0.005036 sec
FRONT   44338   28912   27160   44027   48676   8837    46601   48091   37737   16823
    ...  7  ...
    7   7   7   7   7   7   7   7   7   7

    Doubly Linked List(nodes:100000, show:HEAD/TAIL,10)
```

// On my honor, I pledge that I have neither received nor provided improper assistance in the completion of this assignment.

// Signed: 조다빈 Student Number: 22000711


```
    cpu: 0.005036 sec
FRONT   44338   28912   27160   44027   48676   8837    46601   48091   37737   16823
    ...  7  ...
    7   7   7   7   7   7   7   7   7   7

    Doubly Linked List(nodes:100000, show:HEAD/TAIL,10)
    f - push front       O(1)   p - pop front   O(1)
    b - push back        O(1)   y - pop back    O(1)
    B - push back N      O(n)   Y - pop back N  O(n)
    i - push             O(n)   d - pop         O(n)
    z - push sorted*     O(n)   e - pop all*    O(n)

    s - sorted?          O(n)   r - reverse     O(n)
    x - perfect shuffle* O(n)   a - randomize   O(n)
    u - unique*          O(n)   w - swap pairs  O(n)
    t - show [ALL]          n - n nodes per line
    c - clear       O(n)
    Command[q to quit]: e
    Enter a number to pop all: 7
    cpu: 0.008731 sec
FRONT   44338   28912   27160   44027   48676   8837    46601   48091   37737   16823
    ...  5560  ...
    24854   33993   28952   27273   13348   44403   5564    14628   8604    5052

    Doubly Linked List(nodes:49998, show:HEAD/TAIL,10)
    f - push front       O(1)   p - pop front   O(1)
```

Step4. Half() and show()*

Odd number :

```
    cpu: 4.1e-05 sec
FRONT   1   2   3   4   5

    Doubly Linked List(nodes:5, show:HEAD/TAIL,10)
    f - push front       O(1)   p - pop front   O(1)
    b - push back        O(1)   y - pop back    O(1)
    B - push back N      O(n)   Y - pop back N  O(n)
    i - push             O(n)   d - pop         O(n)
    z - push sorted*     O(n)   e - pop all*    O(n)

    s - sorted?          O(n)   r - reverse     O(n)
    x - perfect shuffle* O(n)   a - randomize   O(n)
    u - unique*          O(n)   w - swap pairs  O(n)
    t - show [ALL]          n - n nodes per line
    c - clear       O(n)
    Command[q to quit]: n
    Enter number of nodes to show per line: 2
FRONT   1   2
    ...  3  ...
    4   5

    Doubly Linked List(nodes:5, show:HEAD/TAIL,2)
    f - push front       O(1)   p - pop front   O(1)
```

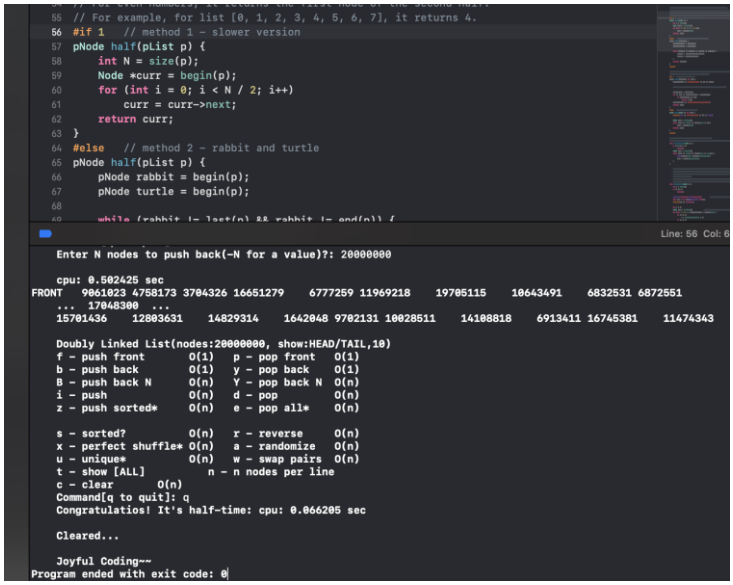
Even number :

```
    Enter a number to push: 6
FRONT   1   2   3   4   5   6

    Doubly Linked List(nodes:6, show:HEAD/TAIL,10)
    f - push front       O(1)   p - pop front   O(1)
    b - push back        O(1)   y - pop back    O(1)
    B - push back N      O(n)   Y - pop back N  O(n)
    i - push             O(n)   d - pop         O(n)
    z - push sorted*     O(n)   e - pop all*    O(n)

    s - sorted?          O(n)   r - reverse     O(n)
    x - perfect shuffle* O(n)   a - randomize   O(n)
    u - unique*          O(n)   w - swap pairs  O(n)
    t - show [ALL]          n - n nodes per line
    c - clear       O(n)
    Command[q to quit]: n
    Enter number of nodes to show per line: 2
FRONT   1   2
    ...  4  ...
    5   6

    Doubly Linked List(nodes:6, show:HEAD/TAIL,2)
    f - push front       O(1)   p - pop front   O(1)
```

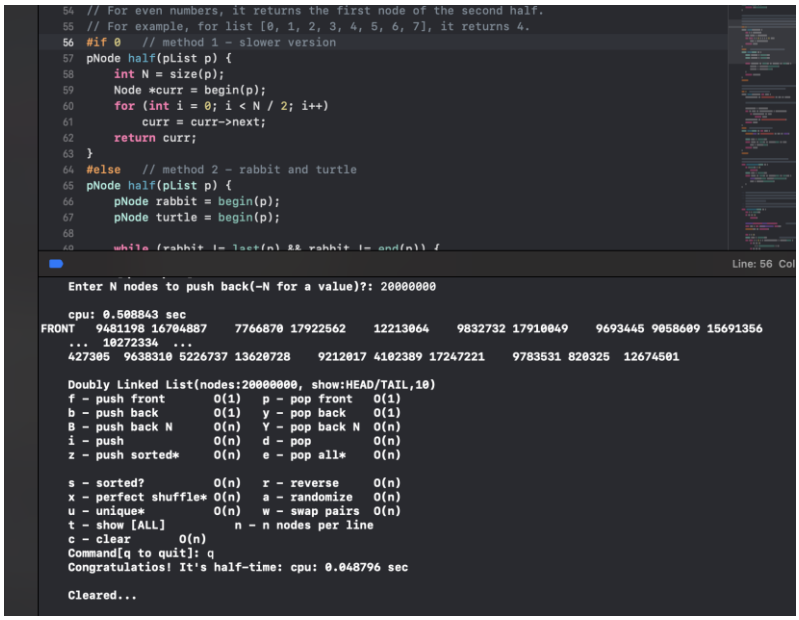
// On my honor, I pledge that I have neither received nor provided improper assistance in the completion of this assignment.

// Signed: 조다빈 Student Number: 22000711

Method 1 :

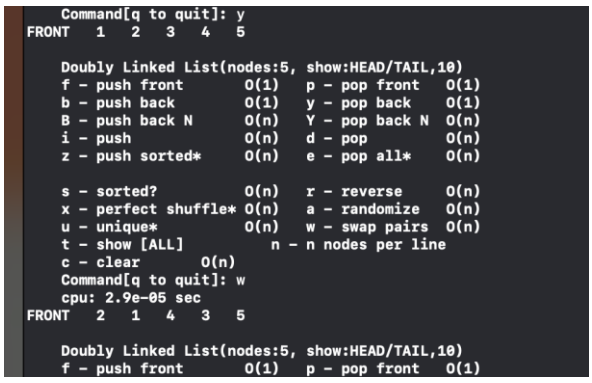


Method 2 :

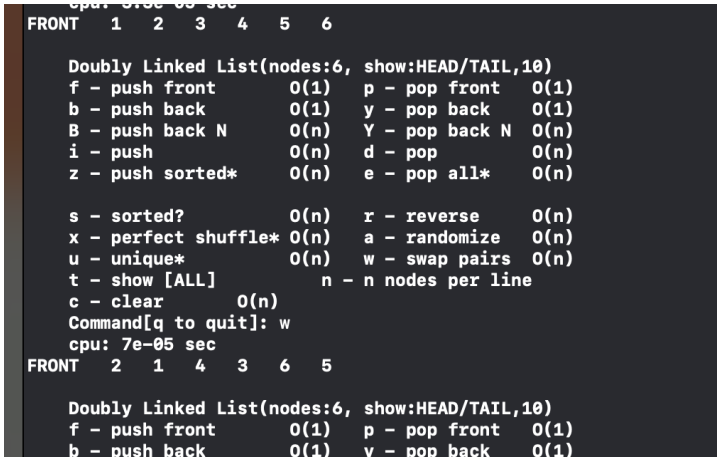


Step5. Swap_pairs

Odd number :

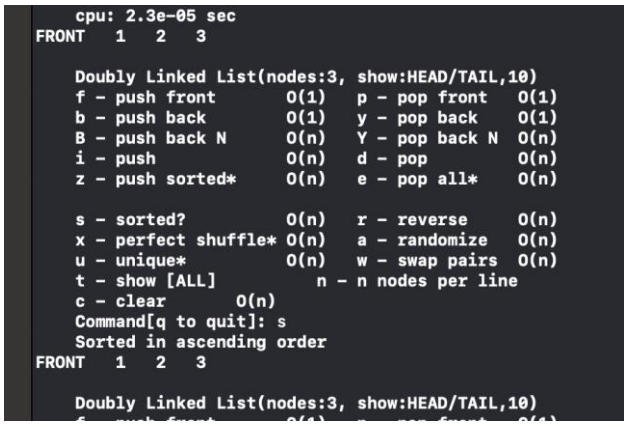


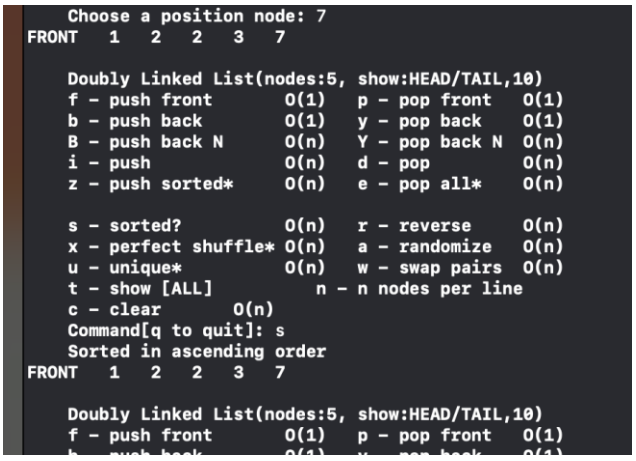
Even number :



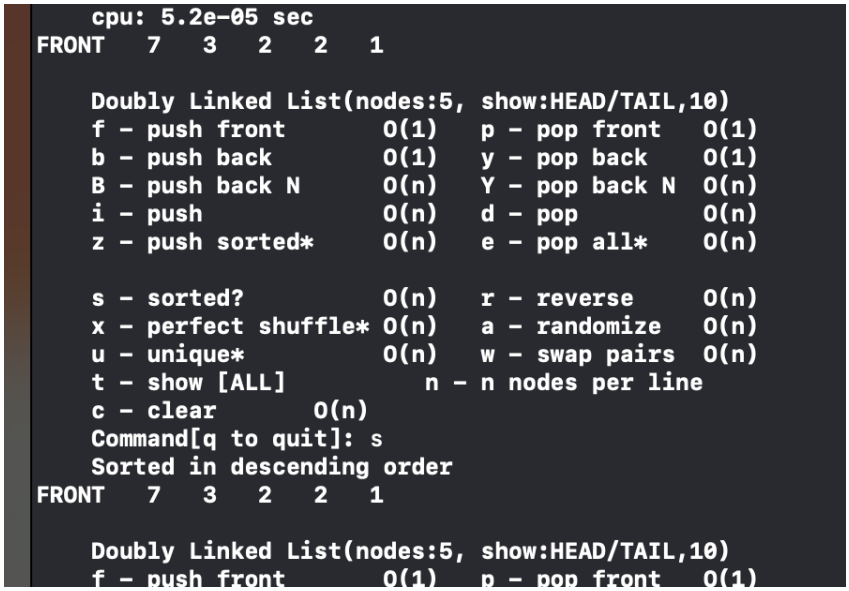
Step6. Sorted()

Ascending :





Descending :



unsorted :

```
    cpu: 3.6e-05 sec
FRONT   0   3   2   5   5   3

    Doubly Linked List(nodes:6, show:HEAD/TAIL,10)
    f - push front       O(1)   p - pop front   O(1)
    b - push back        O(1)   y - pop back    O(1)
    B - push back N      O(n)   Y - pop back N  O(n)
    i - push             O(n)   d - pop         O(n)
    z - push sorted*     O(n)   e - pop all*    O(n)

    s - sorted?          O(n)   r - reverse     O(n)
    x - perfect shuffle* O(n)   a - randomize   O(n)
    u - unique*          O(n)   w - swap pairs  O(n)
    t - show [ALL]          n - n nodes per line
    c - clear       O(n)
    Command[q to quit]: s
    It is unsorted.
FRONT   0   3   2   5   5   3

    Doubly Linked List(nodes:6, show:HEAD/TAIL,10)
    f - push front       O(1)   p - pop front   O(1)
```

Step7. Push_sorted()

Unsorted :

```
    cpu: 5.2e-05 sec
FRONT   9   3   7   5   5   9   9   6

    Doubly Linked List(nodes:8, show:HEAD/TAIL,10)
    f - push front       O(1)   p - pop front   O(1)
    b - push back        O(1)   y - pop back    O(1)
    B - push back N      O(n)   Y - pop back N  O(n)
    i - push             O(n)   d - pop         O(n)
    z - push sorted*     O(n)   e - pop all*    O(n)

    s - sorted?          O(n)   r - reverse     O(n)
    x - perfect shuffle* O(n)   a - randomize   O(n)
    u - unique*          O(n)   w - swap pairs  O(n)
    t - show [ALL]          n - n nodes per line
    c - clear       O(n)
    Command[q to quit]: z
    The operation works in sorted list only.
FRONT   9   3   7   5   5   9   9   6

    Doubly Linked List(nodes:8, show:HEAD/TAIL,10)
    f - push front       O(1)   p - pop front   O(1)
```

Ascending :

```
    cpu: 3.1e-05 sec
FRONT   3   5   5   7   9   9   9

    Doubly Linked List(nodes:7, show:HEAD/TAIL,10)
    f - push front       O(1)   p - pop front   O(1)
    b - push back        O(1)   y - pop back    O(1)
    B - push back N      O(n)   Y - pop back N  O(n)
    i - push             O(n)   d - pop         O(n)
    z - push sorted*     O(n)   e - pop all*    O(n)

    s - sorted?          O(n)   r - reverse     O(n)
    x - perfect shuffle* O(n)   a - randomize   O(n)
    u - unique*          O(n)   w - swap pairs  O(n)
    t - show [ALL]          n - n nodes per line
    c - clear       O(n)
    Command[q to quit]: z
    Enter a number to push: 8
    cpu: 2.8e-05 sec
FRONT   3   5   5   7   8   9   9   9

    Doubly Linked List(nodes:8, show:HEAD/TAIL,10)
    f - push front       O(1)   p - pop front   O(1)
    b - push back        O(1)   y - pop back    O(1)
```

// On my honor, I pledge that I have neither received nor provided improper assistance in the completion of this assignment.

// Signed: 조다빈 Student Number: 22000711

Descending :

```
    cpu: 3.1e-05 sec
FRONT   9   9   9   7   5   5   3

    Doubly Linked List(nodes:7, show:HEAD/TAIL,10)
    f - push front       O(1)   p - pop front   O(1)
    b - push back        O(1)   y - pop back    O(1)
    B - push back N      O(n)   Y - pop back N  O(n)
    i - push             O(n)   d - pop         O(n)
    z - push sorted*     O(n)   e - pop all*    O(n)

    s - sorted?          O(n)   r - reverse     O(n)
    x - perfect shuffle* O(n)   a - randomize   O(n)
    u - unique*          O(n)   w - swap pairs  O(n)
    t - show [ALL]          n - n nodes per line
    c - clear       O(n)
    Command[q to quit]: z
    Enter a number to push: 6
    cpu: 4.6e-05 sec
FRONT   9   9   9   7   6   5   5   3

    Doubly Linked List(nodes:8, show:HEAD/TAIL,10)
    f - push front       O(1)   p - pop front   O(1)
```

Step8. Unique()*

```
    cpu: 3.6e-05 sec
FRONT   1   2   2   3   4   5   5   5   6

    Doubly Linked List(nodes:9, show:HEAD/TAIL,10)
    f - push front       O(1)   p - pop front   O(1)
    b - push back        O(1)   y - pop back    O(1)
    B - push back N      O(n)   Y - pop back N  O(n)
    i - push             O(n)   d - pop         O(n)
    z - push sorted*     O(n)   e - pop all*    O(n)

    s - sorted?          O(n)   r - reverse     O(n)
    x - perfect shuffle* O(n)   a - randomize   O(n)
    u - unique*          O(n)   w - swap pairs  O(n)
    t - show [ALL]          n - n nodes per line
    c - clear       O(n)
    Command[q to quit]: u
    cpu: 3.2e-05 sec
FRONT   1   2   3   4   5   6

    Doubly Linked List(nodes:6, show:HEAD/TAIL,10)
    f - push front       O(1)   p - pop front   O(1)
```

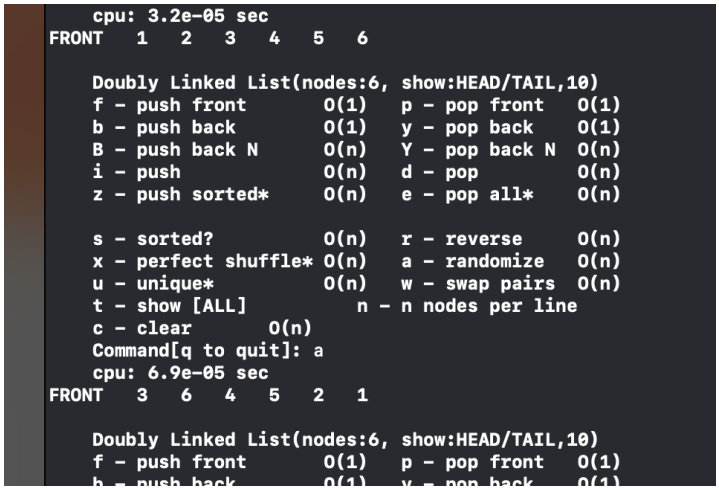
Step9. Reverse()

```
    cpu: 2.4e-05 sec
FRONT   1   2   3   4   5

    Doubly Linked List(nodes:5, show:HEAD/TAIL,10)
    f - push front       O(1)   p - pop front   O(1)
    b - push back        O(1)   y - pop back    O(1)
    B - push back N      O(n)   Y - pop back N  O(n)
    i - push             O(n)   d - pop         O(n)
    z - push sorted*     O(n)   e - pop all*    O(n)

    s - sorted?          O(n)   r - reverse     O(n)
    x - perfect shuffle* O(n)   a - randomize   O(n)
    u - unique*          O(n)   w - swap pairs  O(n)
    t - show [ALL]          n - n nodes per line
    c - clear       O(n)
    Command[q to quit]: r
    cpu: 3.2e-05 sec
FRONT   5   4   3   2   1

    Doubly Linked List(nodes:5, show:HEAD/TAIL,10)
    f - push front       O(1)   p - pop front   O(1)
```

// On my honor, I pledge that I have neither received nor provided improper assistance in the completion of this assignment.

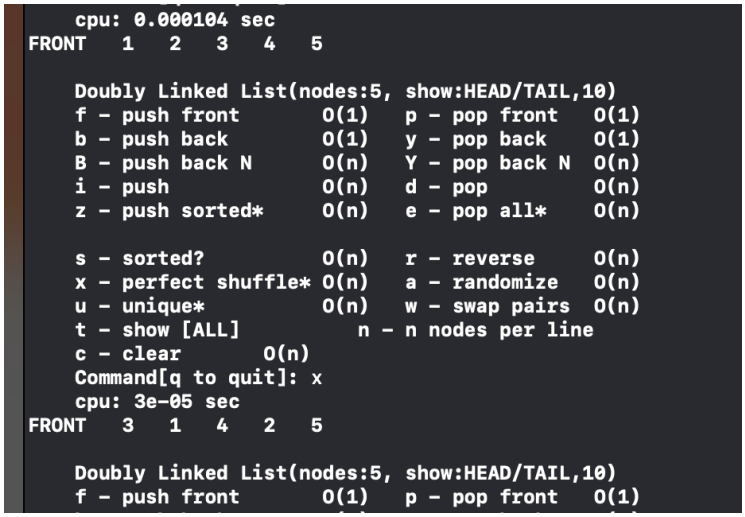
// Signed: 조다빈 Student Number: 22000711

Step10. Randomize()



Step11. Shuffle()*

Odd number :



Even number :