

CIS 22C
Lab 0
Ha Rim Ku (Melody)

Program Outputs

1. Singly Linked List

```
Melodys-MacBook-Air:Lab 0 felons$ python SLinkedList.py
===== Singly Linked List Implementation =====
Operation: Creating List

Current List:
<Empty List>
=====
Operation: Append(1)
Operation: Append(2)
Operation: Append(3)

Current List:
Node 0 : data=3, prev=2
Node 1 : data=2, prev=1
Node 2 : data=1, prev=NULL
=====
Operation: Prepend(0)

Current List:
Node 0 : data=3, prev=2
Node 1 : data=2, prev=1
Node 2 : data=1, prev=0
Node 3 : data=0, prev=NULL
=====
Operation: InsertAfter(1,-1841)

Current List:
Node 0 : data=3, prev=2
Node 1 : data=2, prev=-1841
Node 2 : data=-1841, prev=1
Node 3 : data=1, prev=0
Node 4 : data=0, prev=NULL
=====
Operation: InsertBefore(1,-777)

Current List:
Node 0 : data=3, prev=2
Node 1 : data=2, prev=-1841
Node 2 : data=-1841, prev=1
Node 3 : data=1, prev=-777
Node 4 : data=-777, prev=0
Node 5 : data=0, prev=NULL
=====
Operation: Delete(-777)

Current List:
Node 0 : data=3, prev=2
Node 1 : data=2, prev=-1841
Node 2 : data=-1841, prev=1
```

```
Node 3 : data=1, prev=0
Node 4 : data=0, prev=NULL
```

```
=====
Operation: Search(-777)
There is no such element in the list
Node not Found
```

```
Current List:
Node 0 : data=3, prev=2
Node 1 : data=2, prev=-1841
Node 2 : data=-1841, prev=1
Node 3 : data=1, prev=0
Node 4 : data=0, prev=NULL
```

```
=====
Operation: Search(-1841)
Node found with value -1841
```

```
Operation: Search(99999)
There is no such element in the list
Node not Found
```

```
Current List:
Node 0 : data=3, prev=2
Node 1 : data=2, prev=-1841
Node 2 : data=-1841, prev=1
Node 3 : data=1, prev=0
Node 4 : data=0, prev=NULL
```

```
=====
Operation: IsEmpty()
List is not empty
```

```
Current List:
Node 0 : data=3, prev=2
Node 1 : data=2, prev=-1841
Node 2 : data=-1841, prev=1
Node 3 : data=1, prev=0
Node 4 : data=0, prev=NULL
```

```
=====
Operation: Delete(2)
Operation: Delete(3)
Operation: Delete(-1841)
Operation: Delete(1)
Operation: Delete(0)
```

```
Current List:
<Empty List>
```

```
=====
Operation: IsEmpty()
List is empty
```

```
Current List:
<Empty List>
```

```
=====
Program Finished
Melodys-MacBook-Air:Lab 0 felons$ █
```

2. Doubly Linked List

```
Melodys-MacBook-Air:Lab 0 felons$ python DLinkedList.py
===== Doubly Linked List Implementation =====
Operation: Creating List

Current List:
<Empty List>
=====
Operation: Append(1)
Operation: Append(2)
Operation: Append(3)

Current List:
Node 0 : data=3, prev_data=2, next_data=NULL
Node 1 : data=2, prev_data=1, next_data=3
Node 2 : data=1, prev_data=NULL, next_data=2
=====
Operation: Prepend(0)

Current List:
Node 0 : data=3, prev_data=2, next_data=NULL
Node 1 : data=2, prev_data=1, next_data=3
Node 2 : data=1, prev_data=0, next_data=2
Node 3 : data=0, prev_data=NULL, next_data=1
=====
Operation: InsertAfter(1,-1841)

Current List:
Node 0 : data=3, prev_data=2, next_data=NULL
Node 1 : data=2, prev_data=-1841, next_data=3
Node 2 : data=-1841, prev_data=1, next_data=2
Node 3 : data=1, prev_data=0, next_data=-1841
Node 4 : data=0, prev_data=NULL, next_data=1
=====
Operation: InsertBefore(1,-777)

Current List:
Node 0 : data=3, prev_data=2, next_data=NULL
Node 1 : data=2, prev_data=-1841, next_data=3
Node 2 : data=-1841, prev_data=1, next_data=2
Node 3 : data=1, prev_data=-777, next_data=-1841
Node 4 : data=-777, prev_data=0, next_data=1
Node 5 : data=0, prev_data=NULL, next_data=-777
=====
Operation: Delete(-777)

Current List:
Node 0 : data=3, prev_data=2, next_data=NULL
Node 1 : data=2, prev_data=-1841, next_data=3
Node 2 : data=-1841, prev_data=1, next_data=2
```

Node 3 : data=1, prev_data=0, next_data=-1841
Node 4 : data=0, prev_data=NULL, next_data=1

=====
Operation: Search(-777)
There is no such element in the list
Node not Found

Current List:
Node 0 : data=3, prev_data=2, next_data=NULL
Node 1 : data=2, prev_data=-1841, next_data=3
Node 2 : data=-1841, prev_data=1, next_data=2
Node 3 : data=1, prev_data=0, next_data=-1841
Node 4 : data=0, prev_data=NULL, next_data=1

=====
Operation: Search(-1841)
Node found with value -1841

Operation: Search(99999)
There is no such element in the list
Node not Found

Current List:
Node 0 : data=3, prev_data=2, next_data=NULL
Node 1 : data=2, prev_data=-1841, next_data=3
Node 2 : data=-1841, prev_data=1, next_data=2
Node 3 : data=1, prev_data=0, next_data=-1841
Node 4 : data=0, prev_data=NULL, next_data=1

=====
Operation: IsEmpty()
List is not empty

Current List:
Node 0 : data=3, prev_data=2, next_data=NULL
Node 1 : data=2, prev_data=-1841, next_data=3
Node 2 : data=-1841, prev_data=1, next_data=2
Node 3 : data=1, prev_data=0, next_data=-1841
Node 4 : data=0, prev_data=NULL, next_data=1

=====
Operation: Delete(2)
Operation: Delete(3)
Operation: Delete(-1841)
Operation: Delete(1)
Operation: Delete(0)

Current List:
<Empty List>

=====
Operation: IsEmpty()
List is empty

Current List:
<Empty List>

=====
Program Finished

Melodys-MacBook-Air:Lab 0 felons\$ █

3. Stack

```
Melodys-MacBook-Air:Lab 0 felons$ python Stack.py
===== Stack Implementation =====
Operation: Creating Stack

Current Stack:
<Empty Stack>
=====
Operation: push(0)
Operation: push(1)
Operation: push(2)
Operation: push(3)

Current Stack:
Node 0 : data=3, prev=2, next=NULL
Node 1 : data=2, prev=1, next=3
Node 2 : data=1, prev=0, next=2
Node 3 : data=0, prev=NULL, next=1
=====
Operation: getLength()
Length of current Stack: 4

Current Stack:
Node 0 : data=3, prev=2, next=NULL
Node 1 : data=2, prev=1, next=3
Node 2 : data=1, prev=0, next=2
Node 3 : data=0, prev=NULL, next=1
=====
Operation: pop()

Current Stack:
Node 0 : data=2, prev=1, next=NULL
Node 1 : data=1, prev=0, next=2
Node 2 : data=0, prev=NULL, next=1
=====
Operation: peek()
Item at the top of stack: 2

Current Stack:
Node 0 : data=2, prev=1, next=NULL
Node 1 : data=1, prev=0, next=2
Node 2 : data=0, prev=NULL, next=1
=====
Operation: isEmpty()
Stack is not empty

Current Stack:
Node 0 : data=2, prev=1, next=NULL
Node 1 : data=1, prev=0, next=2
Node 2 : data=0, prev=NULL, next=1
```

```
=====
Operation: pop()
```

```
Current Stack:
```

```
Node 0 : data=1, prev=0, next=NULL
```

```
Node 1 : data=0, prev=NULL, next=1
```

```
=====
Operation: pop()
```

```
Current Stack:
```

```
Node 0 : data=0, prev=NULL, next=NULL
```

```
=====
Operation: pop()
```

```
Current Stack:
```

```
<Empty Stack>
```

```
=====
Operation: isEmpty()
```

```
Stack is empty
```

```
Current Stack:
```

```
<Empty Stack>
```

```
=====
Operation: getLength()
```

```
Length of current Stack: 0
```

```
Current Stack:
```

```
<Empty Stack>
```

```
=====
Program Finished
```

```
Melodys-MacBook-Air:Lab 0 felons$
```

4. Queue

```
Melodys-MacBook-Air:Lab 0 felons$ python Queue.py
===== Queue Implementation =====
Operation: Creating Queue

Current Queue:
<Empty Queue>
=====
Operation: push(0)
Operation: push(1)
Operation: push(2)
Operation: push(3)

Current Queue:
Node 0 : data=3, prev=2, next=NULL
Node 1 : data=2, prev=1, next=3
Node 2 : data=1, prev=0, next=2
Node 3 : data=0, prev=NULL, next=1
=====
Operation: getLength()
Length of current Queue: 4

Current Queue:
Node 0 : data=3, prev=2, next=NULL
Node 1 : data=2, prev=1, next=3
Node 2 : data=1, prev=0, next=2
Node 3 : data=0, prev=NULL, next=1
=====
Operation: pop()

Current Queue:
Node 0 : data=3, prev=2, next=NULL
Node 1 : data=2, prev=1, next=3
Node 2 : data=1, prev=NULL, next=2
=====
Operation: peek()
Item at the front of Queue: 1

Current Queue:
Node 0 : data=3, prev=2, next=NULL
Node 1 : data=2, prev=1, next=3
Node 2 : data=1, prev=NULL, next=2
=====
Operation: isEmpty()
Queue is not empty

Current Queue:
Node 0 : data=3, prev=2, next=NULL
Node 1 : data=2, prev=1, next=3
Node 2 : data=1, prev=NULL, next=2
```

```
=====
Operation: isEmpty()
Queue is not empty
```

```
Current Queue:
Node 0 : data=3, prev=2, next=NULL
Node 1 : data=2, prev=1, next=3
Node 2 : data=1, prev=NULL, next=2
=====
```

```
Operation: pop()
```

```
Current Queue:
Node 0 : data=3, prev=2, next=NULL
Node 1 : data=2, prev=NULL, next=3
=====
```

```
Operation: pop()
```

```
Current Queue:
Node 0 : data=3, prev=NULL, next=NULL
=====
```

```
Operation: pop()
```

```
Current Queue:
<Empty Queue>
=====
```

```
Operation: isEmpty()
Queue is empty
```

```
Current Queue:
<Empty Queue>
=====
```

```
Operation: getLength()
Length of current Queue: 0
```

```
Current Queue:
<Empty Queue>
=====
```

```
Program Finished
Melodys-MacBook-Air:Lab 0 felons$
```


References

“Find Length of a Linked List (Iterative and Recursive).” *GeeksforGeeks*, 1 May 2019, www.geeksforgeeks.org/find-length-of-a-linked-list-iterative-and-recursive/.

Malik, Usman. “Doubly Linked List with Python Examples.” *Stack Abuse*, Stack Abuse, stackabuse.com/doubly-linked-list-with-python-examples/.

Python - Advanced Linked List.

www.tutorialspoint.com/python_data_structure/python_advanced_linked_list.htm.

Shiver, John, and John Shiver John is a Seattle-based web developer with a focus in Python and Django. As a recent graduate of the Code Fellows Python Development Accelerator.

“Implementing a Singly Linked List in Python.” *Code Fellows*, 10 Sept. 2014, www.codefellows.org/blog/implementing-a-singly-linked-list-in-python/.