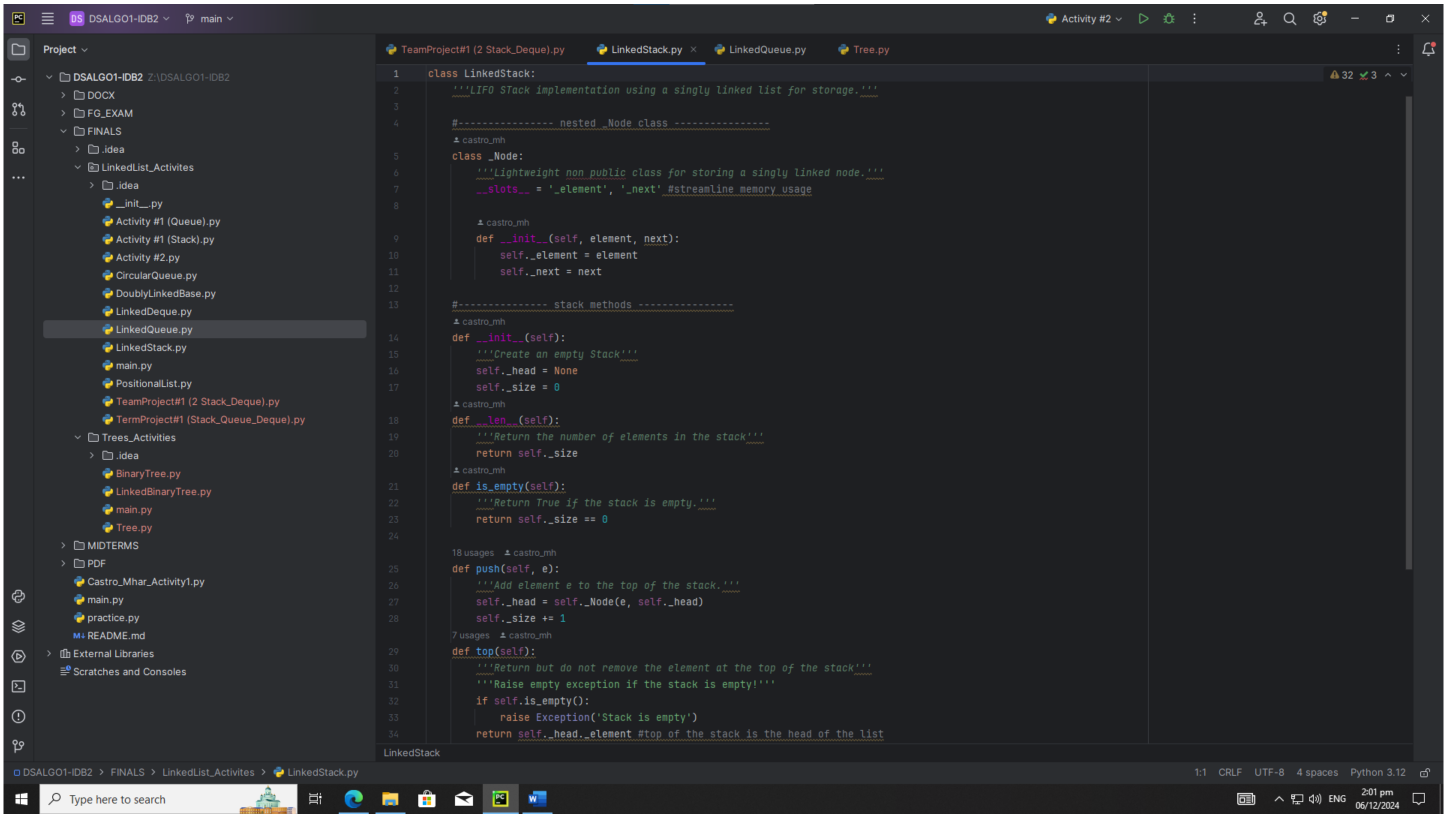
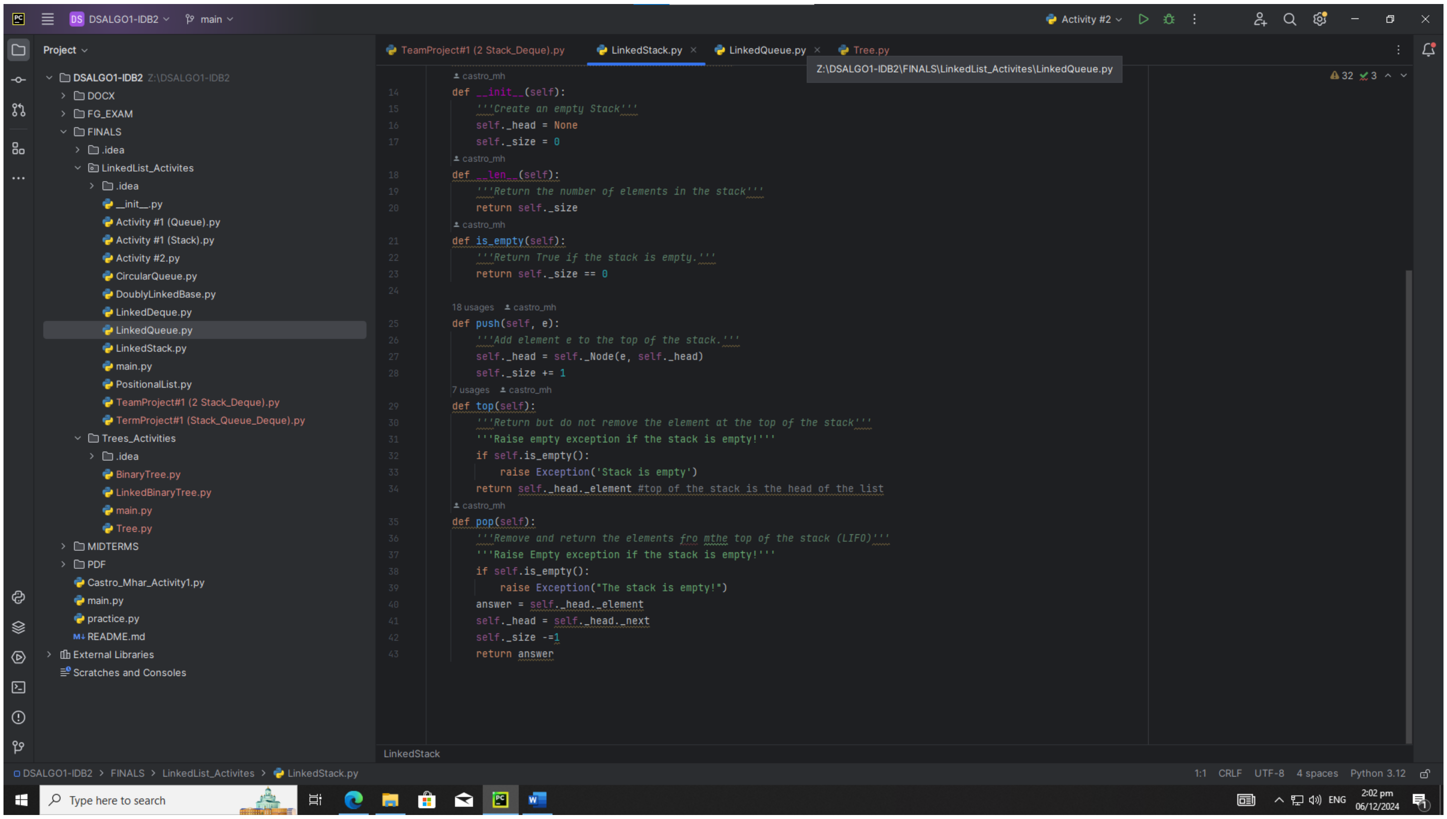
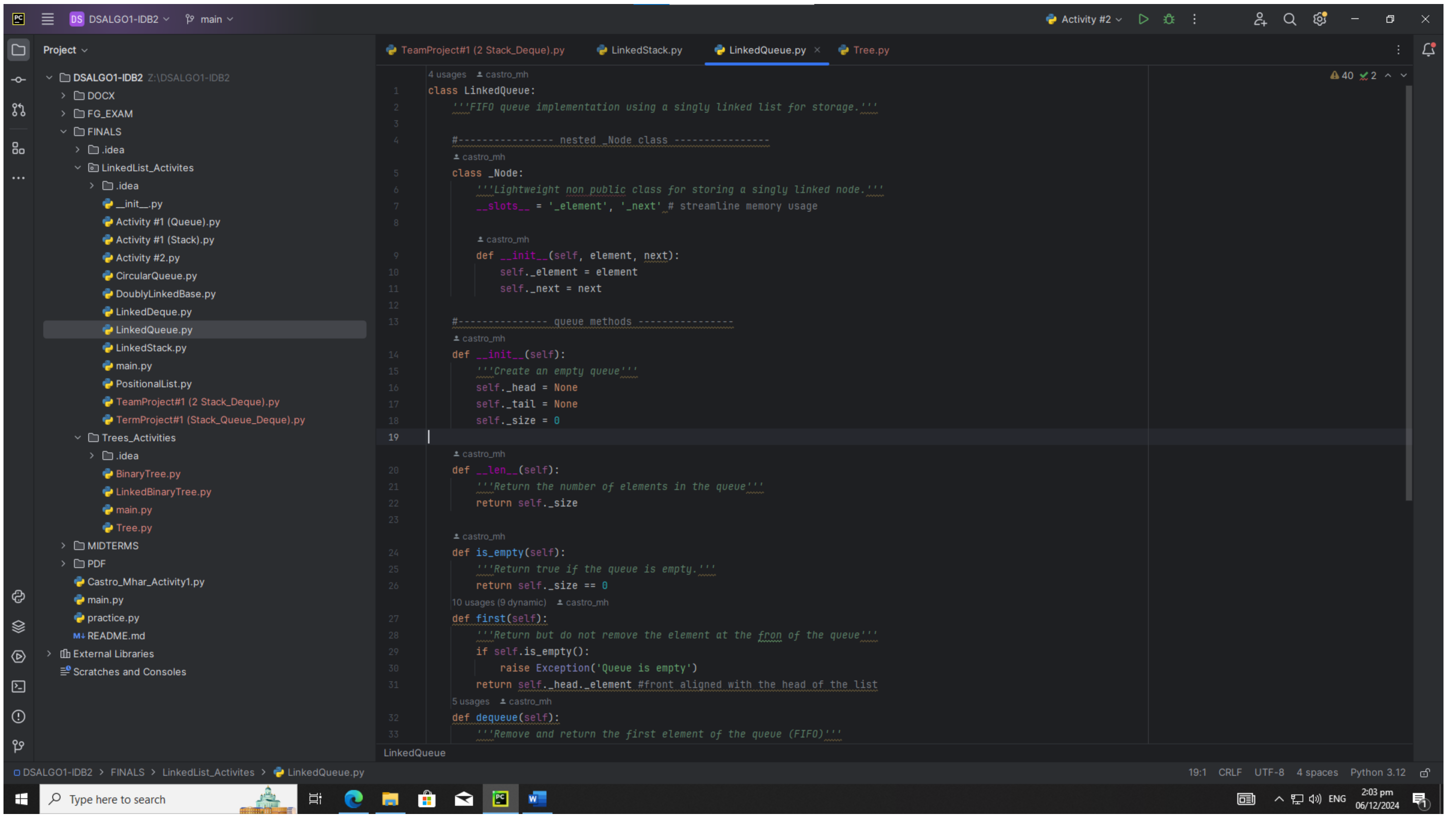
SOURCE CODE:


TeamProject#1 (2 Stack_Deque).py
LinkedStack.py
LinkedQueue.py
Tree.py
DSALGO1-IDB2 > FINALS > LinkedList_Activites > LinkedStack.py

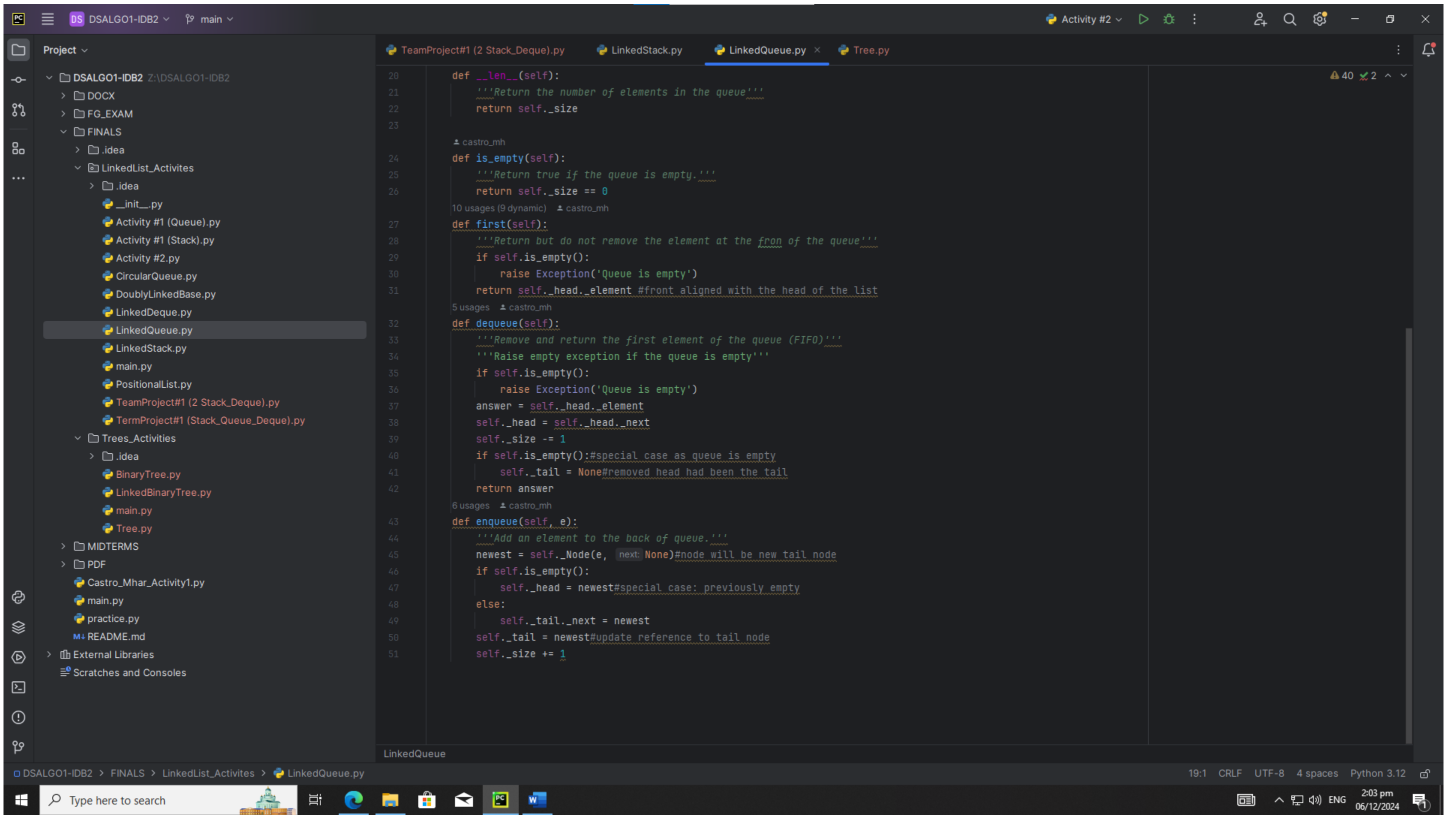
```python
class LinkedStack:
    '''LIFO STack implementation using a singly linked list for storage.'''

    #--------------- nested _Node class ---------------
    class _Node:
        '''Lightweight non public class for storing a singly linked node.'''
        __slots__ = '_element', '_next' #streamline memory usage

        def __init__(self, element, next):
            self._element = element
            self._next = next

    #-------------- stack methods ---------------
    def __init__(self):
        '''Create an empty Stack'''
        self._head = None
        self._size = 0
    def __len__(self):
        '''Return the number of elements in the stack'''
        return self._size
    def is_empty(self):
        '''Return True if the stack is empty.'''
        return self._size == 0

    def push(self, e):
        '''Add element e to the top of the stack.'''
        self._head = self._Node(e, self._head)
        self._size += 1
    def top(self):
        '''Return but do not remove the element at the top of the stack'''
        '''Raise empty exception if the stack is empty!'''
        if self.is_empty():
            raise Exception('Stack is empty')
        return self._head._element #top of the stack is the head of the list
```


TeamProject#1 (2 Stack_Deque).py
LinkedStack.py
LinkedQueue.py
Tree.py
Z:\DSALGO1-IDB2\FINALS\LinkedList_Activites\LinkedQueue.py
DSALGO1-IDB2 > FINALS > LinkedList_Activites > LinkedStack.py

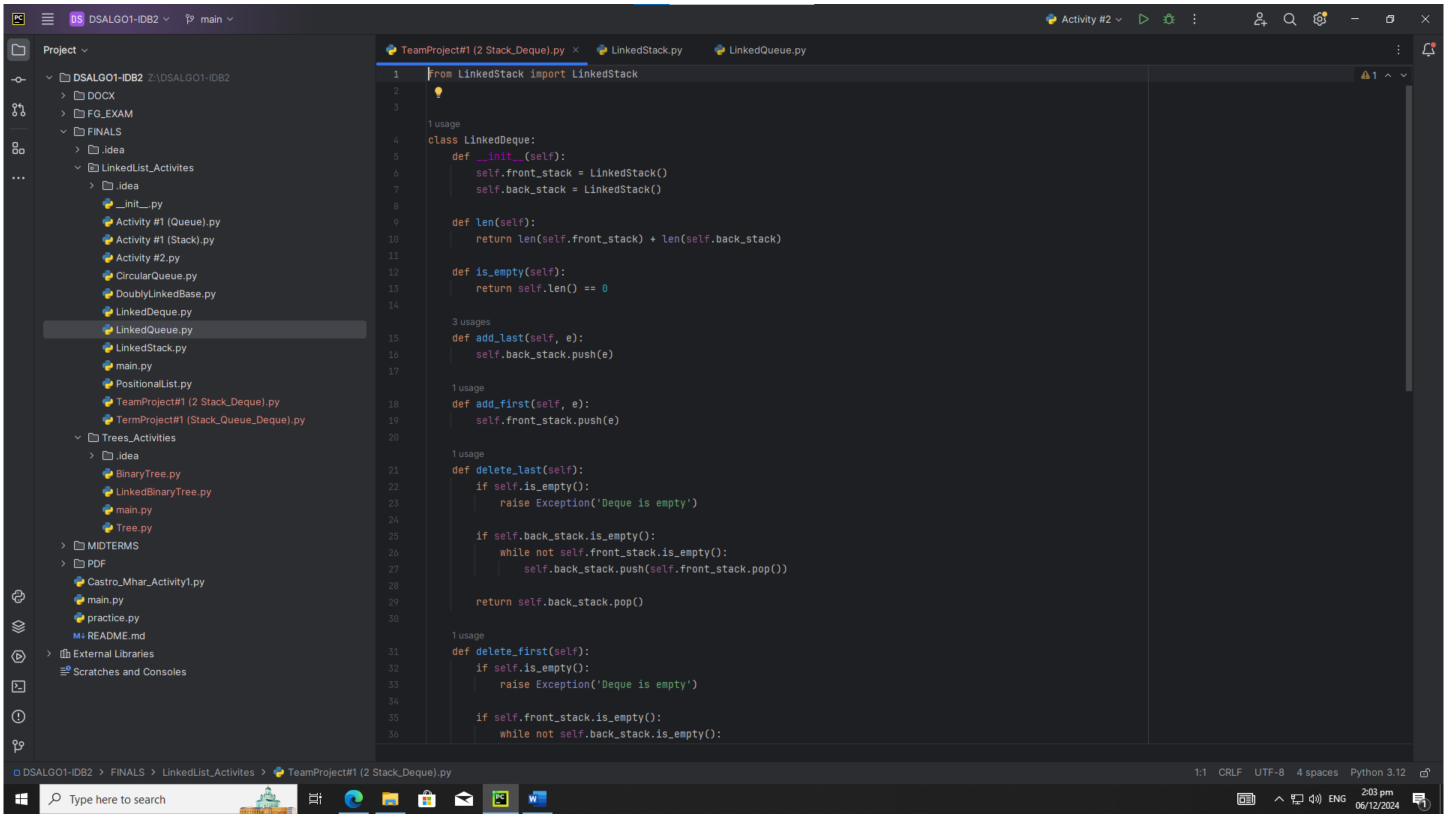
```python
    def __init__(self):
        '''Create an empty Stack'''
        self._head = None
        self._size = 0
    def __len__(self):
        '''Return the number of elements in the stack'''
        return self._size
    def is_empty(self):
        '''Return True if the stack is empty.'''
        return self._size == 0

    def push(self, e):
        '''Add element e to the top of the stack.'''
        self._head = self._Node(e, self._head)
        self._size += 1
    def top(self):
        '''Return but do not remove the element at the top of the stack'''
        '''Raise empty exception if the stack is empty!'''
        if self.is_empty():
            raise Exception('Stack is empty')
        return self._head._element #top of the stack is the head of the list
    def pop(self):
        '''Remove and return the elements fro mthe top of the stack (LIFO)'''
        '''Raise Empty exception if the stack is empty!'''
        if self.is_empty():
            raise Exception("The stack is empty!")
        answer = self._head._element
        self._head = self._head._next
        self._size -=1
        return answer
```



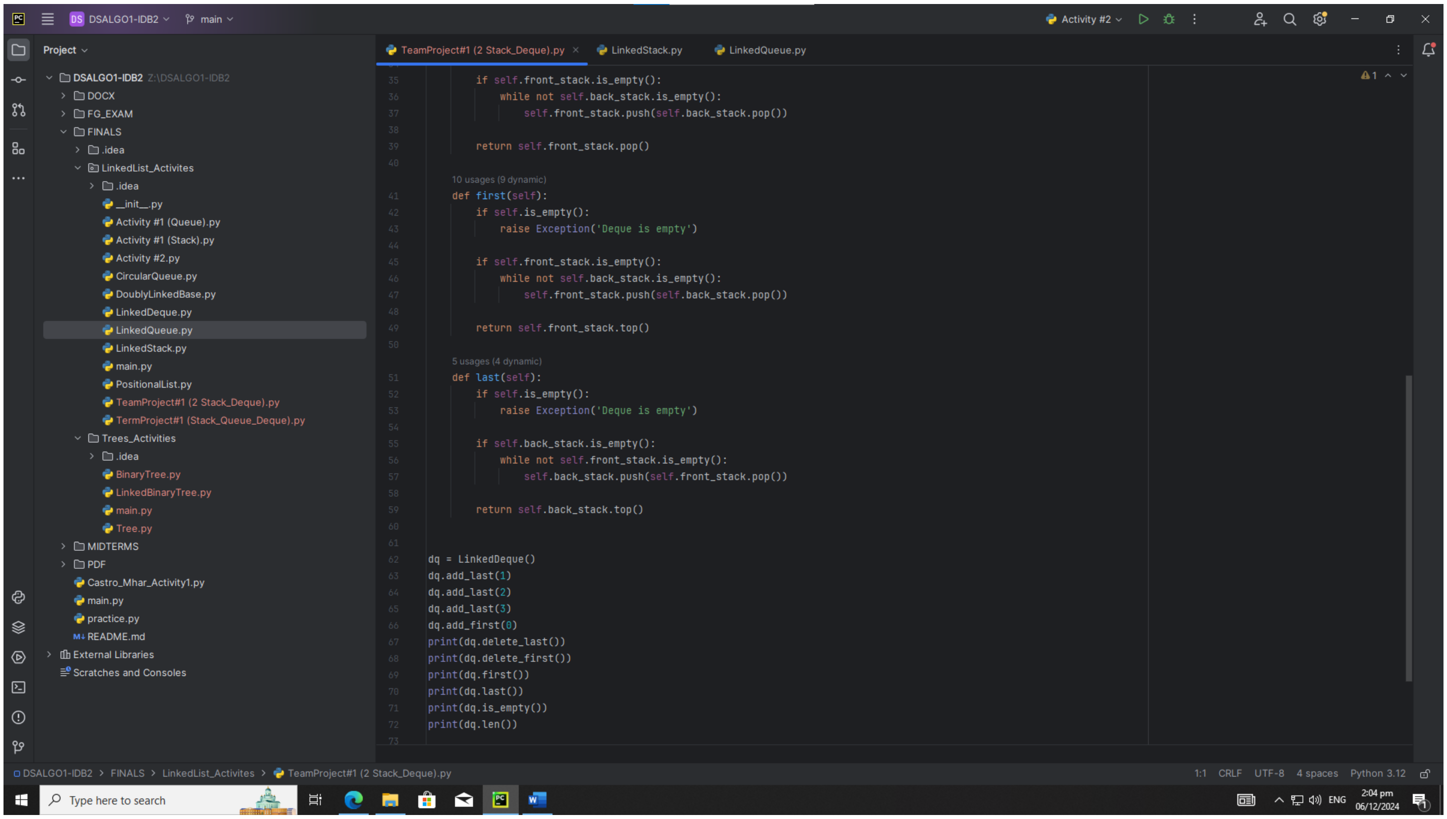
```python
class LinkedQueue:
    '''FIFO queue implementation using a singly linked list for storage.'''

    #---------------- nested _Node class ----------------
    class _Node:
        '''Lightweight non public class for storing a singly linked node.'''
        __slots__ = '_element', '_next' # streamline memory usage

        def __init__(self, element, next):
            self._element = element
            self._next = next

    #--------------- queue methods ----------------
    def __init__(self):
        '''Create an empty queue'''
        self._head = None
        self._tail = None
        self._size = 0

    def __len__(self):
        '''Return the number of elements in the queue'''
        return self._size

    def is_empty(self):
        '''Return true if the queue is empty.'''
        return self._size == 0
    def first(self):
        '''Return but do not remove the element at the fron of the queue'''
        if self.is_empty():
            raise Exception('Queue is empty')
        return self._head._element #front aligned with the head of the list
    def dequeue(self):
        '''Remove and return the first element of the queue (FIFO)'''
```


DSALGO1-IDB2
main
Activity #2
Project
TeamProject#1 (2 Stack_Deque).py
LinkedStack.py
LinkedQueue.py
Tree.py
DSALGO1-IDB2 > FINALS > LinkedList_Activites > LinkedQueue.py
19:1 CRLF UTF-8 4 spaces Python 3.12
Type here to search
ENG
2:03 pm
06/12/2024

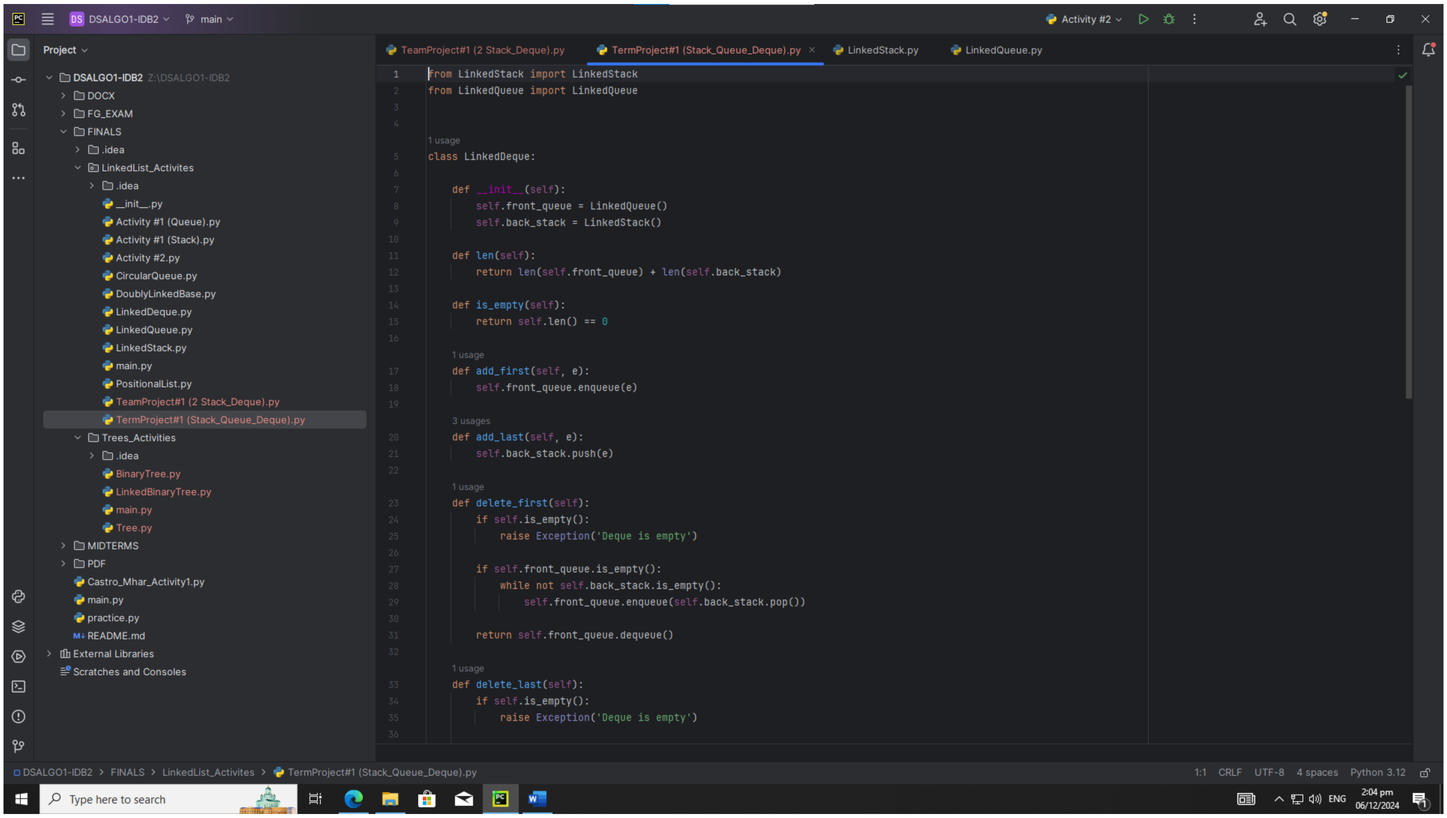
```python
    def __len__(self):
        '''Return the number of elements in the queue'''
        return self._size

    def is_empty(self):
        '''Return true if the queue is empty.'''
        return self._size == 0
    def first(self):
        '''Return but do not remove the element at the fron of the queue'''
        if self.is_empty():
            raise Exception('Queue is empty')
        return self._head._element #front aligned with the head of the list
    def dequeue(self):
        '''Remove and return the first element of the queue (FIFO)'''
        '''Raise empty exception if the queue is empty'''
        if self.is_empty():
            raise Exception('Queue is empty')
        answer = self._head._element
        self._head = self._head._next
        self._size -= 1
        if self.is_empty():#special case as queue is empty
            self._tail = None#removed head had been the tail
        return answer
    def enqueue(self, e):
        '''Add an element to the back of queue.'''
        newest = self._Node(e, next: None)#node will be new tail node
        if self.is_empty():
            self._head = newest#special case: previously empty
        else:
            self._tail._next = newest
        self._tail = newest#update reference to tail node
        self._size += 1
```


TeamProject#1 (2 Stack_Deque).py
LinkedStack.py
LinkedQueue.py
DSALGO1-IDB2 > FINALS > LinkedList_Activites > TeamProject#1 (2 Stack_Deque).py

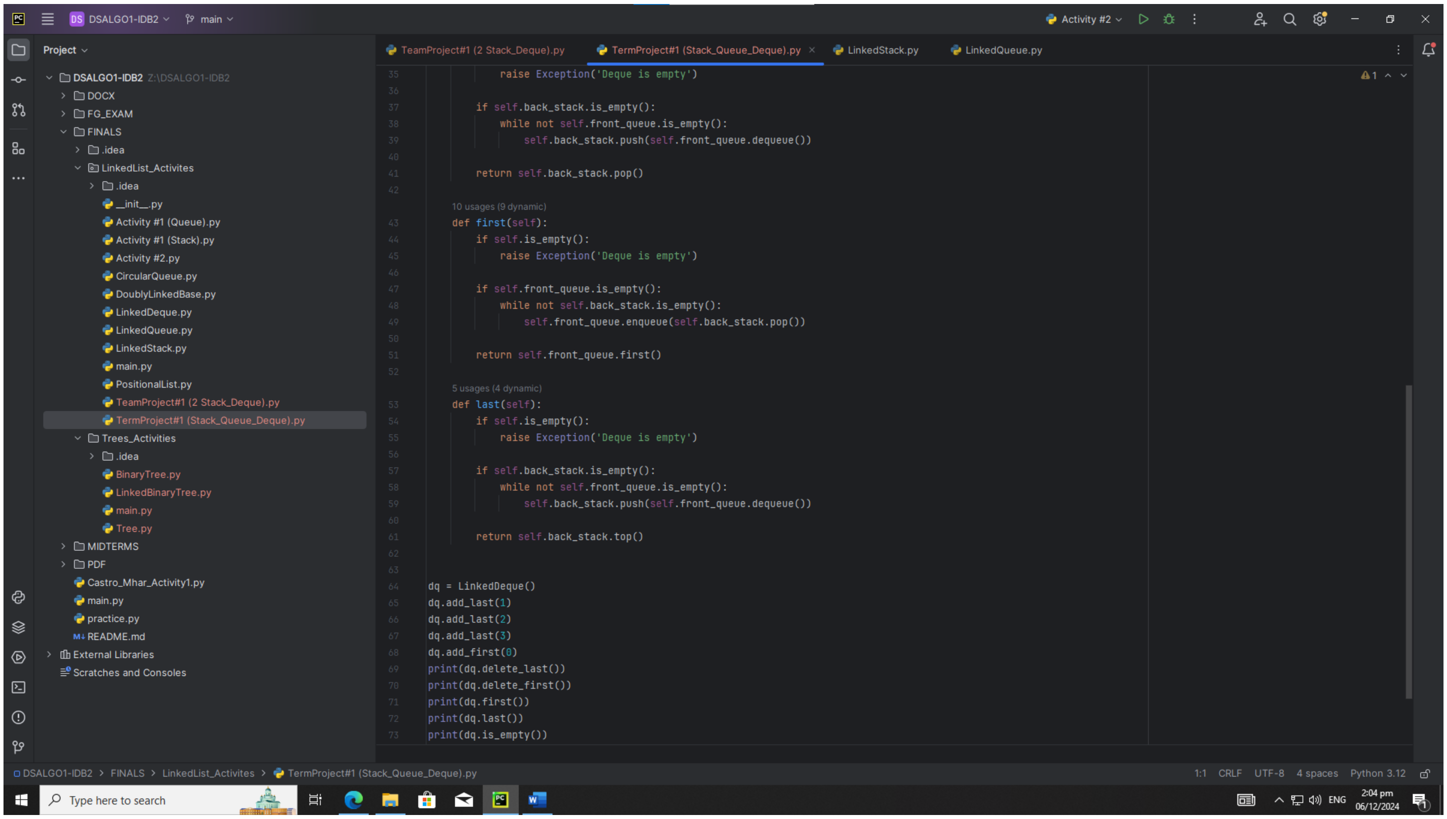
```python
from LinkedStack import LinkedStack


class LinkedDeque:
    def __init__(self):
        self.front_stack = LinkedStack()
        self.back_stack = LinkedStack()

    def len(self):
        return len(self.front_stack) + len(self.back_stack)

    def is_empty(self):
        return self.len() == 0

    def add_last(self, e):
        self.back_stack.push(e)

    def add_first(self, e):
        self.front_stack.push(e)

    def delete_last(self):
        if self.is_empty():
            raise Exception('Deque is empty')

        if self.back_stack.is_empty():
            while not self.front_stack.is_empty():
                self.back_stack.push(self.front_stack.pop())

        return self.back_stack.pop()

    def delete_first(self):
        if self.is_empty():
            raise Exception('Deque is empty')

        if self.front_stack.is_empty():
            while not self.back_stack.is_empty():
```


TeamProject#1 (2 Stack_Deque).py
LinkedStack.py
LinkedQueue.py
DSALGO1-IDB2 > FINALS > LinkedList_Activites > TeamProject#1 (2 Stack_Deque).py

```python
        if self.front_stack.is_empty():
            while not self.back_stack.is_empty():
                self.front_stack.push(self.back_stack.pop())

        return self.front_stack.pop()

    def first(self):
        if self.is_empty():
            raise Exception('Deque is empty')

        if self.front_stack.is_empty():
            while not self.back_stack.is_empty():
                self.front_stack.push(self.back_stack.pop())

        return self.front_stack.top()

    def last(self):
        if self.is_empty():
            raise Exception('Deque is empty')

        if self.back_stack.is_empty():
            while not self.front_stack.is_empty():
                self.back_stack.push(self.front_stack.pop())

        return self.back_stack.top()


dq = LinkedDeque()
dq.add_last(1)
dq.add_last(2)
dq.add_last(3)
dq.add_first(0)
print(dq.delete_last())
print(dq.delete_first())
print(dq.first())
print(dq.last())
print(dq.is_empty())
print(dq.len())
```

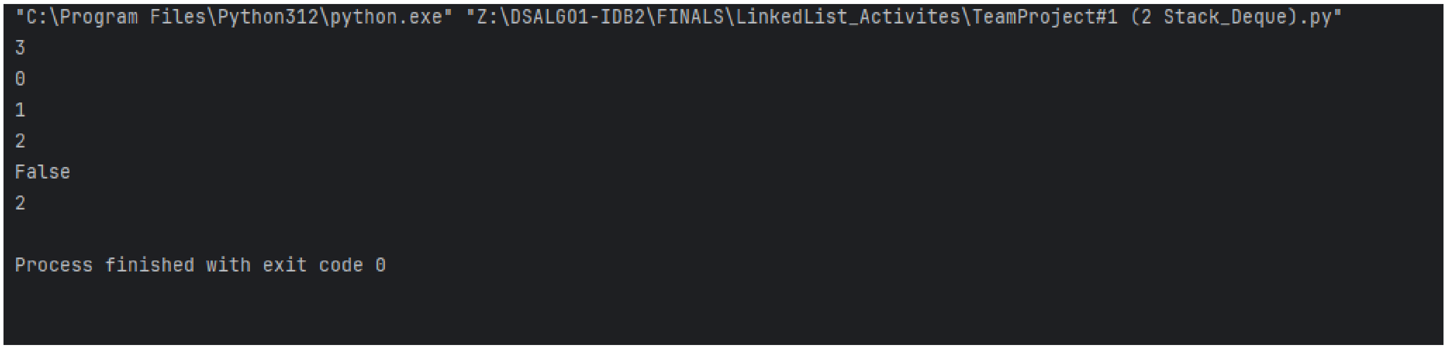

DSALGO1-IDB2
main
Activity #2
Project
TeamProject#1 (2 Stack_Deque).py
TermProject#1 (Stack_Queue_Deque).py
LinkedStack.py
LinkedQueue.py
DSALGO1-IDB2 > FINALS > LinkedList_Activites > TermProject#1 (Stack_Queue_Deque).py
1:1 CRLF UTF-8 4 spaces Python 3.12
2:04 pm
06/12/2024

```python
from LinkedStack import LinkedStack
from LinkedQueue import LinkedQueue


class LinkedDeque:

    def __init__(self):
        self.front_queue = LinkedQueue()
        self.back_stack = LinkedStack()

    def len(self):
        return len(self.front_queue) + len(self.back_stack)

    def is_empty(self):
        return self.len() == 0

    def add_first(self, e):
        self.front_queue.enqueue(e)

    def add_last(self, e):
        self.back_stack.push(e)

    def delete_first(self):
        if self.is_empty():
            raise Exception('Deque is empty')

        if self.front_queue.is_empty():
            while not self.back_stack.is_empty():
                self.front_queue.enqueue(self.back_stack.pop())

        return self.front_queue.dequeue()

    def delete_last(self):
        if self.is_empty():
            raise Exception('Deque is empty')

```


DSALGO1-IDB2
main
Activity #2
Project
TeamProject#1 (2 Stack_Deque).py
TermProject#1 (Stack_Queue_Deque).py
LinkedStack.py
LinkedQueue.py
DSALGO1-IDB2 > FINALS > LinkedList_Activites > TermProject#1 (Stack_Queue_Deque).py
1:1 CRLF UTF-8 4 spaces Python 3.12
2:04 pm
06/12/2024

```python
            raise Exception('Deque is empty')

        if self.back_stack.is_empty():
            while not self.front_queue.is_empty():
                self.back_stack.push(self.front_queue.dequeue())

        return self.back_stack.pop()

    def first(self):
        if self.is_empty():
            raise Exception('Deque is empty')

        if self.front_queue.is_empty():
            while not self.back_stack.is_empty():
                self.front_queue.enqueue(self.back_stack.pop())

        return self.front_queue.first()

    def last(self):
        if self.is_empty():
            raise Exception('Deque is empty')

        if self.back_stack.is_empty():
            while not self.front_queue.is_empty():
                self.back_stack.push(self.front_queue.dequeue())

        return self.back_stack.top()


dq = LinkedDeque()
dq.add_last(1)
dq.add_last(2)
dq.add_last(3)
dq.add_first(0)
print(dq.delete_last())
print(dq.delete_first())
print(dq.first())
print(dq.last())
print(dq.is_empty())
```

OUTPUT:



```
"C:\Program Files\Python312\python.exe" "Z:\DSALGO1-IDB2\FINALS\LinkedList_Activites\TeamProject#1 (2 Stack_Deque).py"
3
0
1
2
False
2

Process finished with exit code 0
```