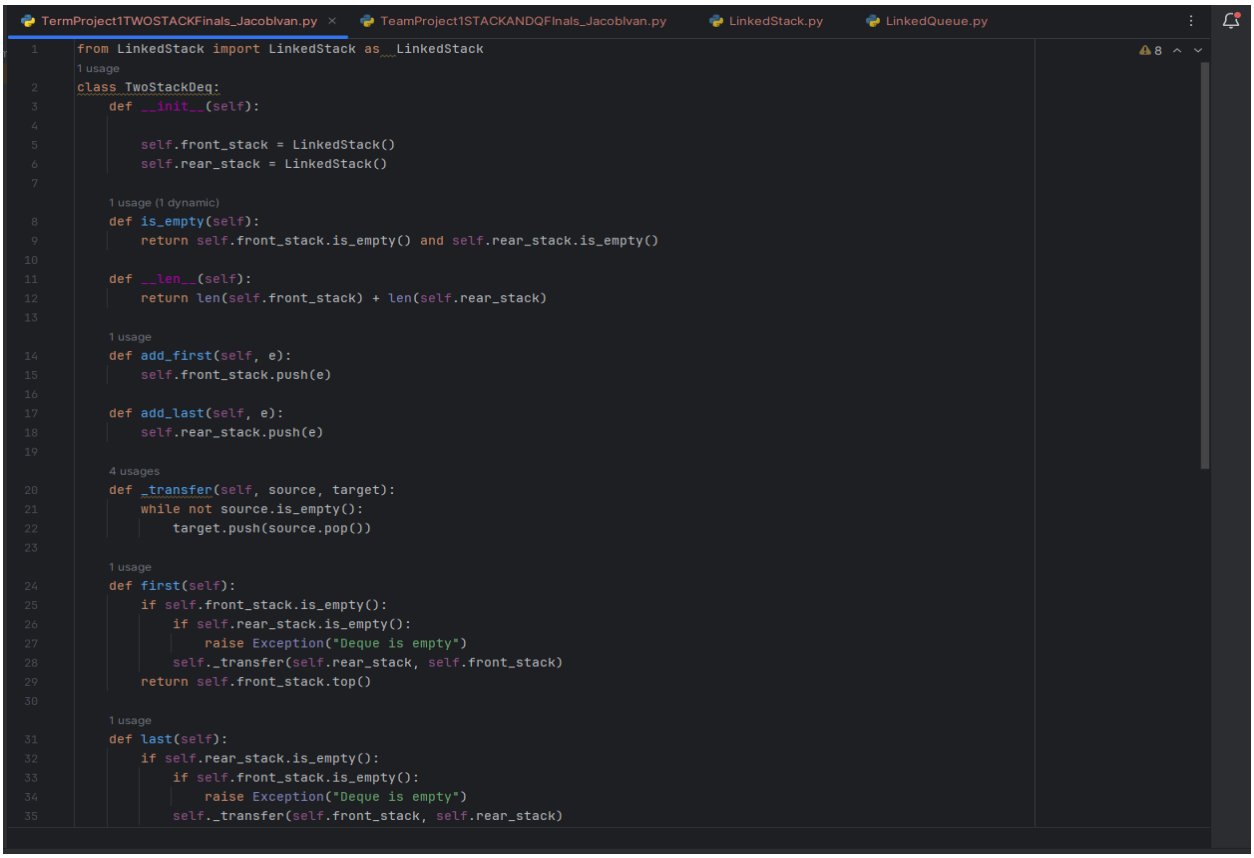
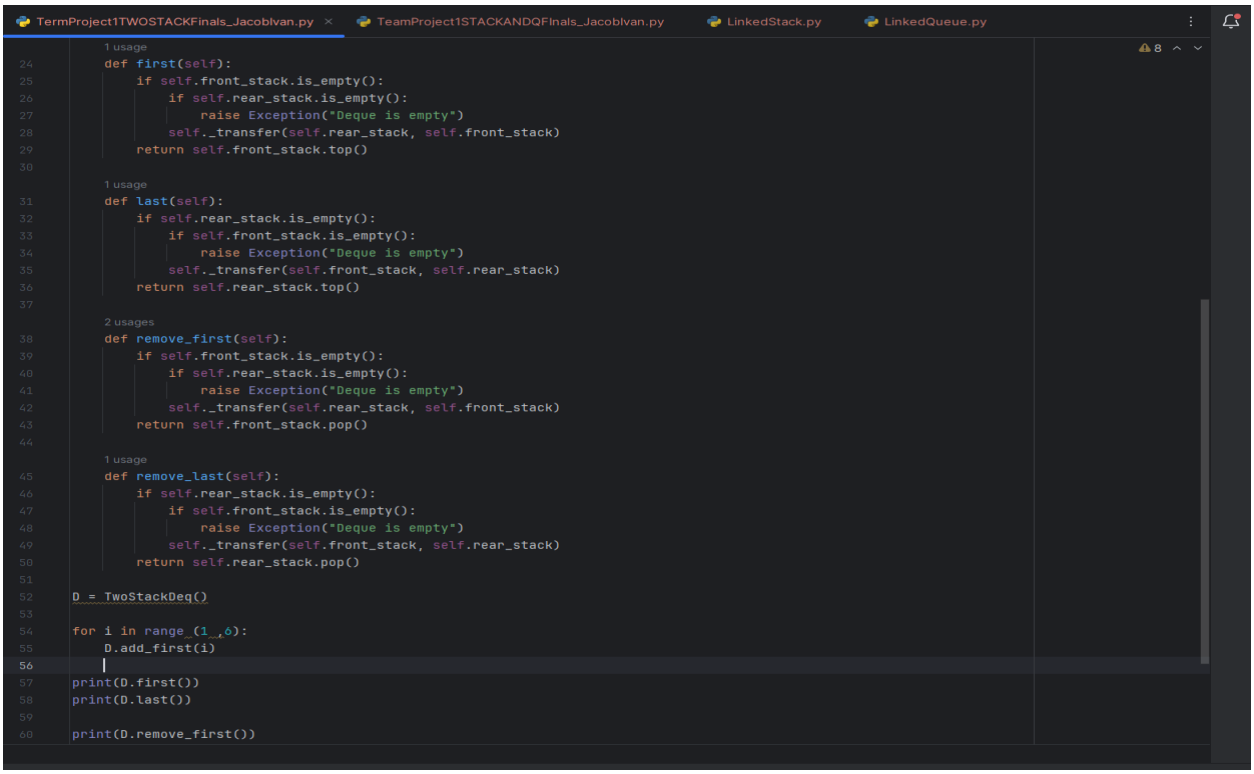
# 2 STACKS


TermProject1TWOSTACKFinals_Jacoblvan.py
TeamProject1STACKANDQFInals_Jacoblvan.py
LinkedStack.py
LinkedQueue.py
8
from LinkedStack import LinkedStack as _LinkedStack
1 usage
class TwoStackDeq:
def __init__(self):
self.front_stack = LinkedStack()
self.rear_stack = LinkedStack()
1 usage (1 dynamic)
def is_empty(self):
return self.front_stack.is_empty() and self.rear_stack.is_empty()
def __len__(self):
return len(self.front_stack) + len(self.rear_stack)
1 usage
def add_first(self, e):
self.front_stack.push(e)
def add_last(self, e):
self.rear_stack.push(e)
4 usages
def _transfer(self, source, target):
while not source.is_empty():
target.push(source.pop())
1 usage
def first(self):
if self.front_stack.is_empty():
if self.rear_stack.is_empty():
raise Exception("Deque is empty")
self._transfer(self.rear_stack, self.front_stack)
return self.front_stack.top()
1 usage
def last(self):
if self.rear_stack.is_empty():
if self.front_stack.is_empty():
raise Exception("Deque is empty")
self._transfer(self.front_stack, self.rear_stack)


TermProject1TWOSTACKFinals_Jacoblvan.py
TeamProject1STACKANDQFInals_Jacoblvan.py
LinkedStack.py
LinkedQueue.py
8
1 usage
def first(self):
if self.front_stack.is_empty():
if self.rear_stack.is_empty():
raise Exception("Deque is empty")
self._transfer(self.rear_stack, self.front_stack)
return self.front_stack.top()
1 usage
def last(self):
if self.rear_stack.is_empty():
if self.front_stack.is_empty():
raise Exception("Deque is empty")
self._transfer(self.front_stack, self.rear_stack)
return self.rear_stack.top()
2 usages
def remove_first(self):
if self.front_stack.is_empty():
if self.rear_stack.is_empty():
raise Exception("Deque is empty")
self._transfer(self.rear_stack, self.front_stack)
return self.front_stack.pop()
1 usage
def remove_last(self):
if self.rear_stack.is_empty():
if self.front_stack.is_empty():
raise Exception("Deque is empty")
self._transfer(self.front_stack, self.rear_stack)
return self.rear_stack.pop()
D = TwoStackDeq()
for i in range (1, 6):
D.add_first(i)
print(D.first())
print(D.last())
print(D.remove_first())

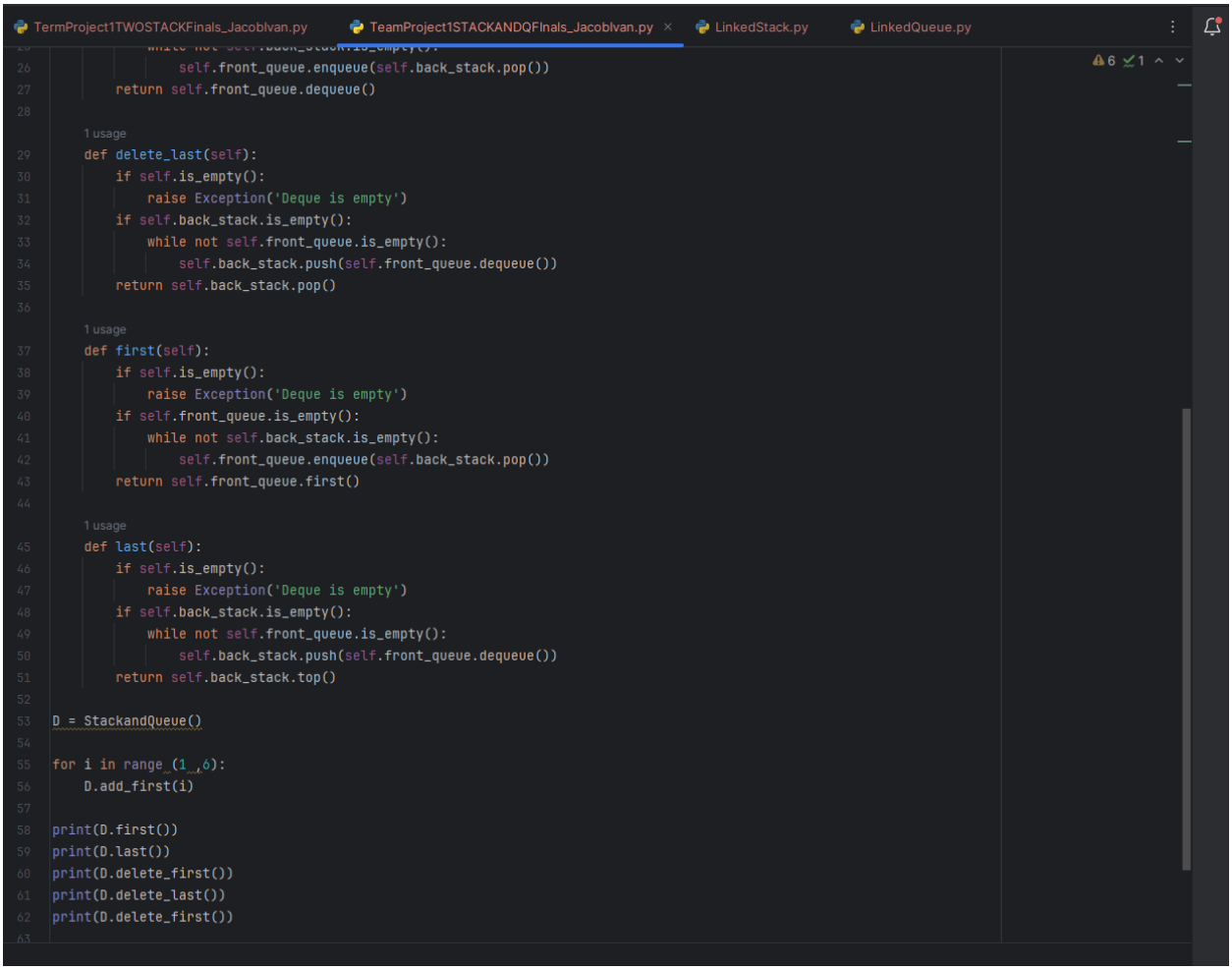
Run TermProject1TWOSTACKFinals_JacobIvan

```
"C:\Program Files\Python312\python.exe" Z:\DSALG01-IDB2\FINALS\TermProjectFinals_JacobIvan\TermProject1TWOSTACKFinals_JacobIvan.py
5
1
5
1
4
```

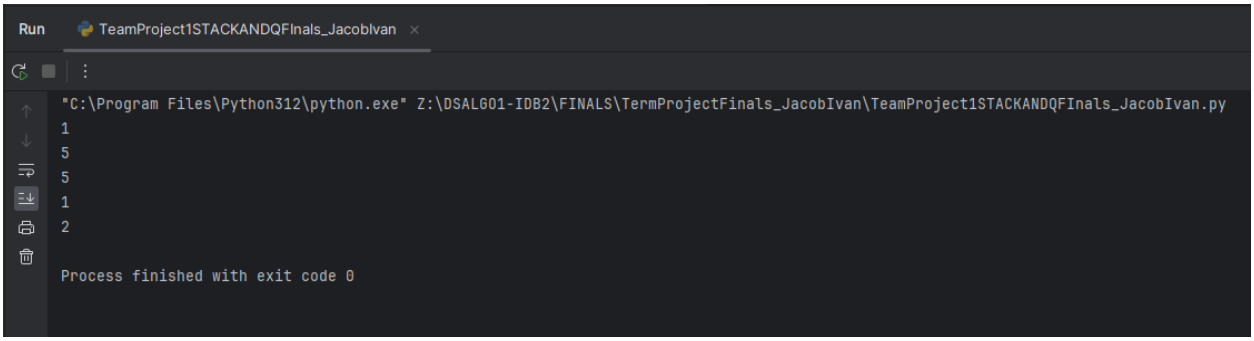
STACK AND QUEUES

TermProject1TWOSTACKFinals_JacobIvan.py | TeamProject1STACKANDQFInals_JacobIvan.py | LinkedStack.py | LinkedQueue.py

```python
from LinkedStack import LinkedStack
from LinkedQueue import LinkedQueue
1 usage
class StackandQueue:

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

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

    5 usages (1 dynamic)
    def is_empty(self):
        return self.len() == 0

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

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

    2 usages
    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()

    1 usage
    def delete_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.pop()
```



```
TermProject1TWOSTACKFinals_JacobIvan.py   TeamProject1STACKANDQFinals_JacobIvan.py   LinkedStack.py   LinkedQueue.py

                self.front_queue.enqueue(self.back_stack.pop())
        return self.front_queue.dequeue()

    1 usage
    def delete_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.pop()

    1 usage
    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()

    1 usage
    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()

D = StackandQueue()

for i in range (1 ,6):
    D.add_first(i)

print(D.first())
print(D.last())
print(D.delete_first())
print(D.delete_last())
print(D.delete_first())
```



```
Run   TeamProject1STACKANDQFinals_JacobIvan

"C:\Program Files\Python312\python.exe" Z:\DSALGO1-IDB2\FINALS\TermProjectFinals_JacobIvan\TeamProject1STACKANDQFinals_JacobIvan.py
1
5
5
1
2

Process finished with exit code 0
```