AI Assignment:11.2

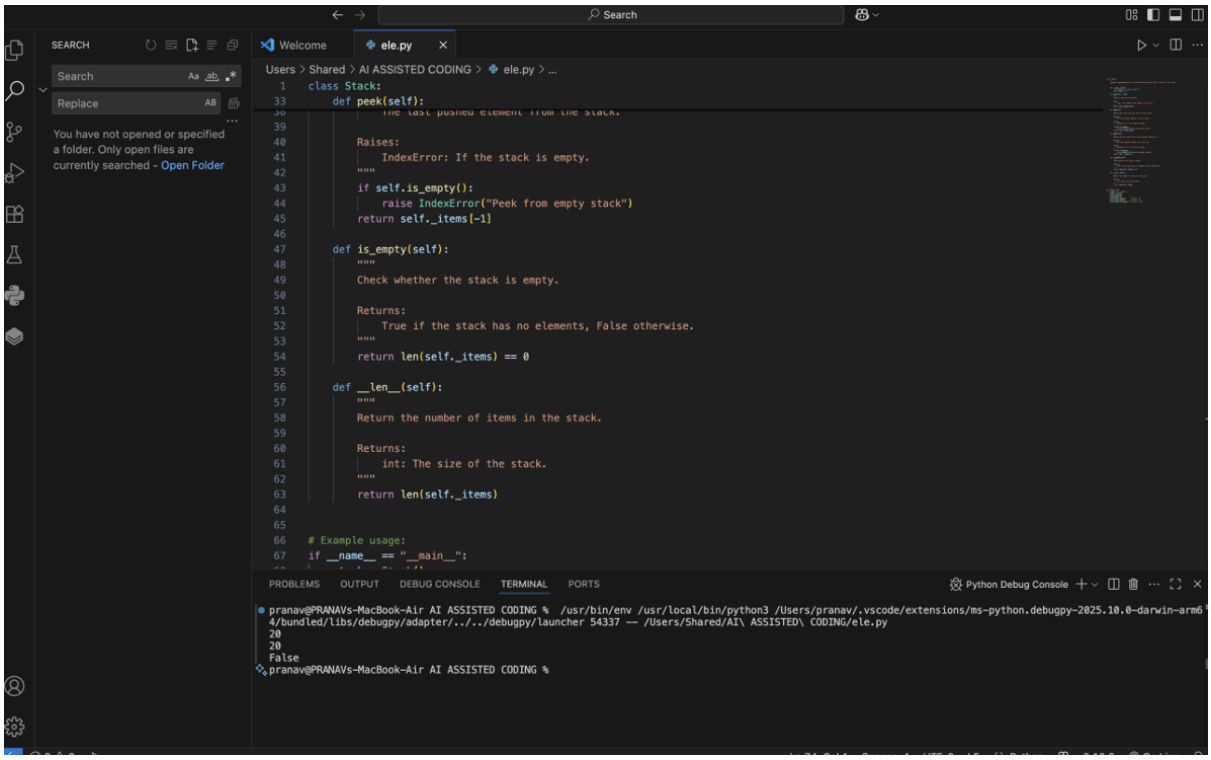
Rollno:2403A51102

Name:balaji

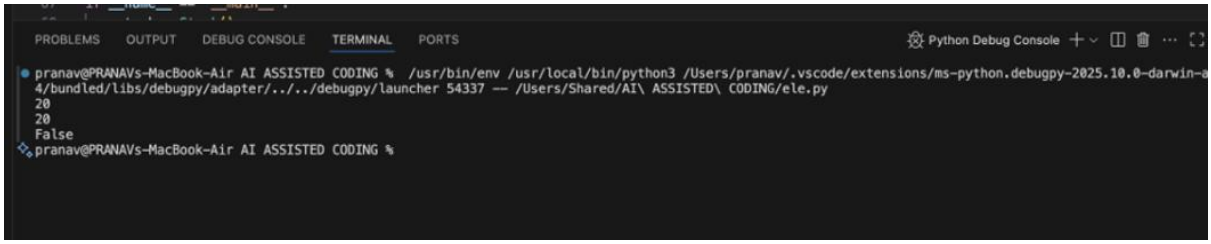
Batch:06

Task Description #1 – Stack Implementation Task: Use AI to generate a Stack class with push, pop, peek, and is_empty methods. Sample Input Code: class Stack: pass Expected Output: • A functional stack implementation with all required methods and docstrings.

Code:



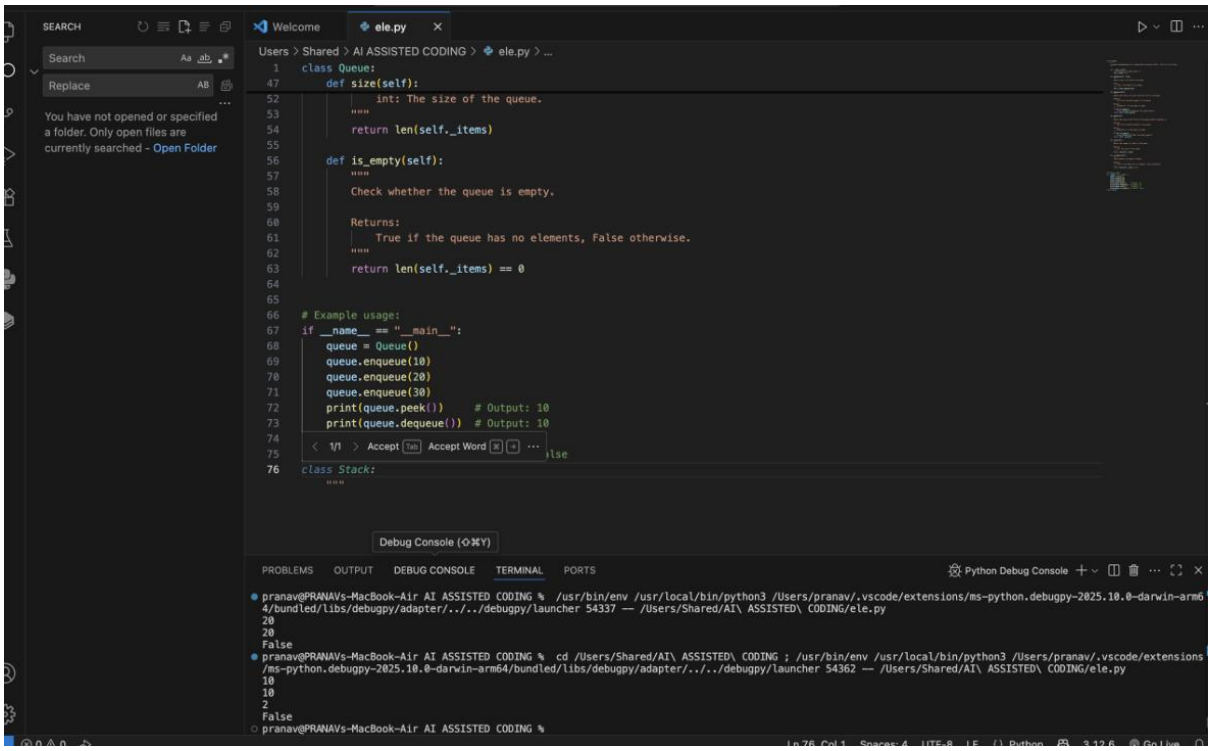
Output:



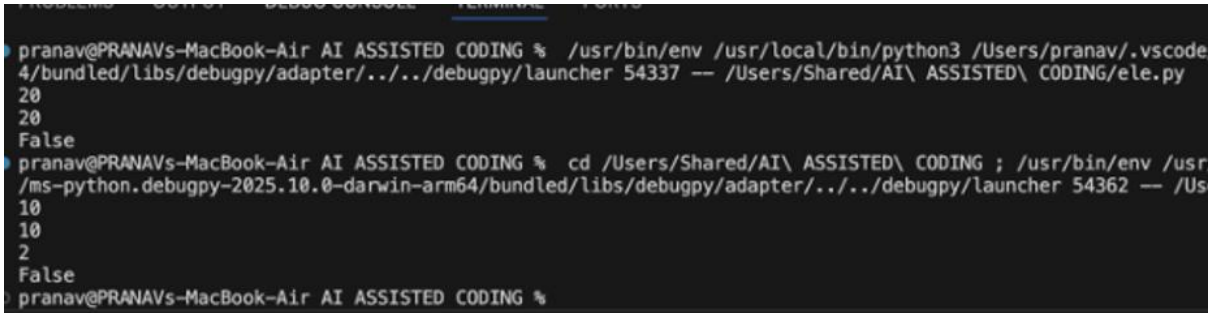
PROMPT : To generate a Stack class with push, pop, peek, and is_empty methods.
Task Description #2 – Queue Implementation Task: Use AI to implement a Queue using Python lists. Sample Input Code: class Queue: pass Expected Output:

- FIFO-based queue class with enqueue, dequeue, peek, and size methods

Code:



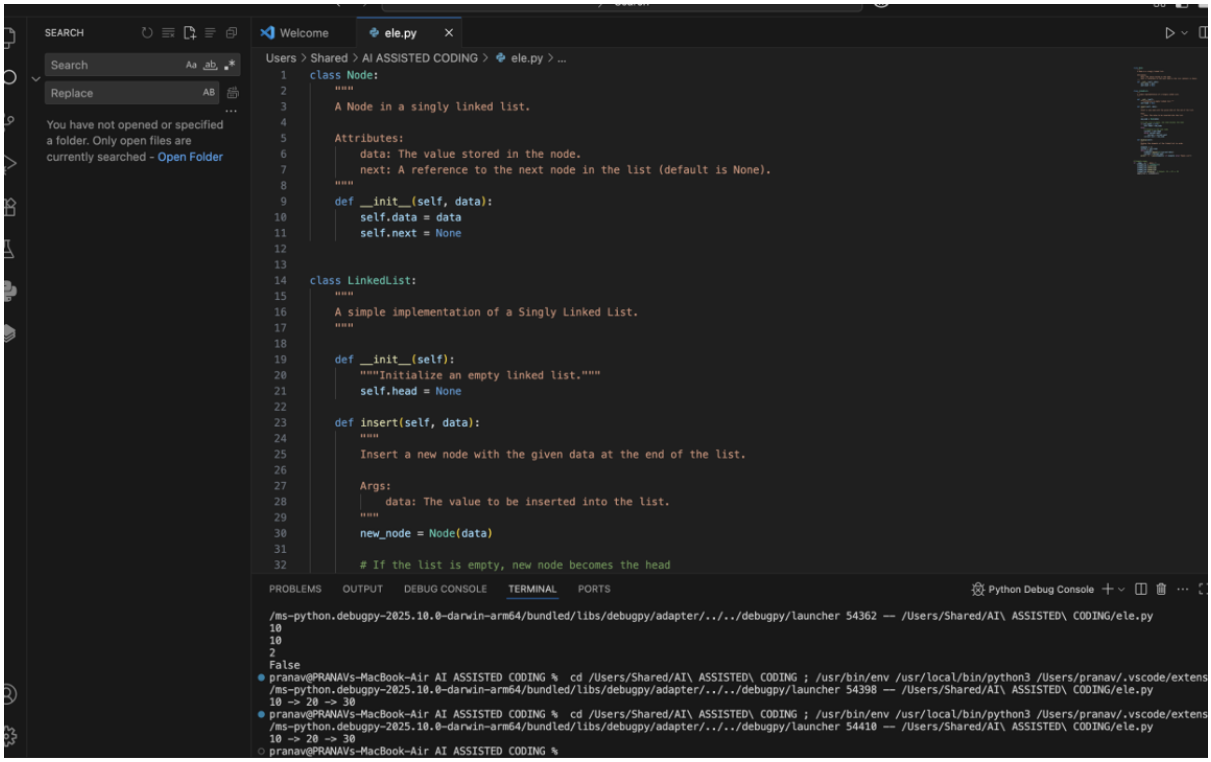
Output:

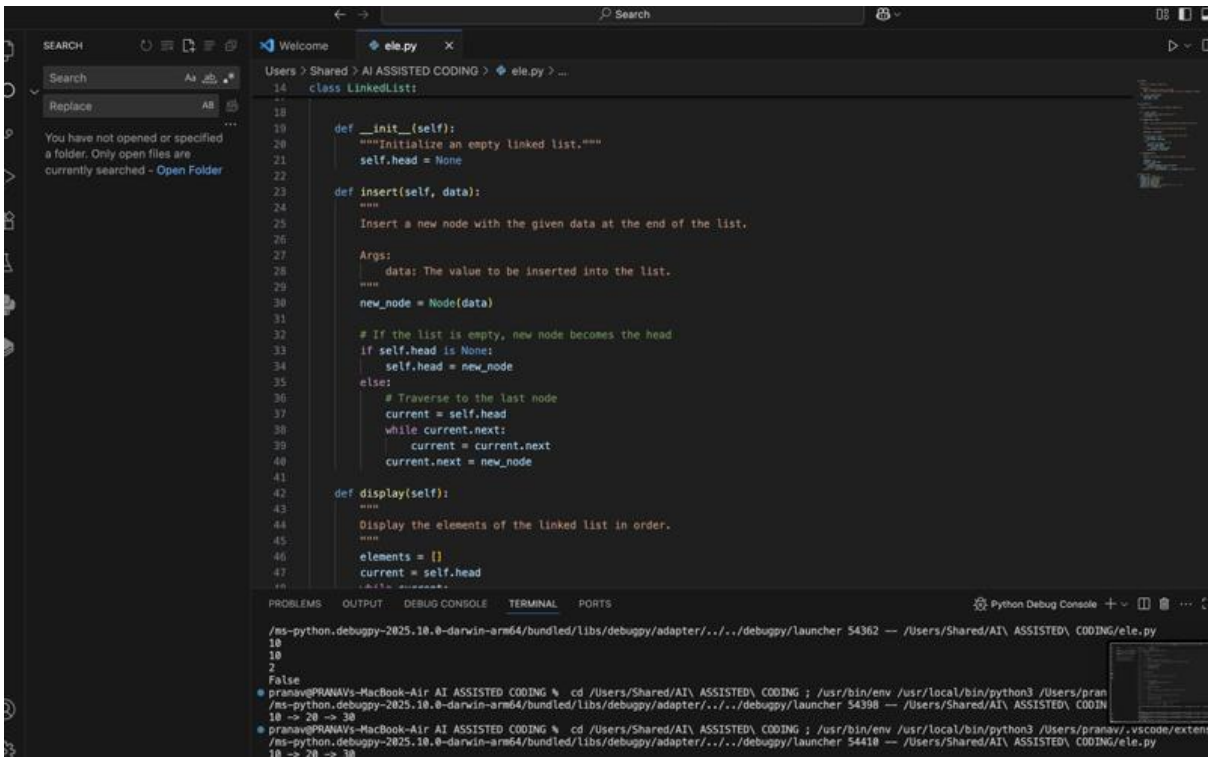


PROMPT : To implement a Queue using Python lists in a simple and basic way Task Description #3 – Linked List Task: Use AI to generate a Singly Linked List with insert and display methods. Sample Input Code: class Node:

pass class LinkedList: pass Expected Output: • A working linked list implementation with clear method documentation.

Code:



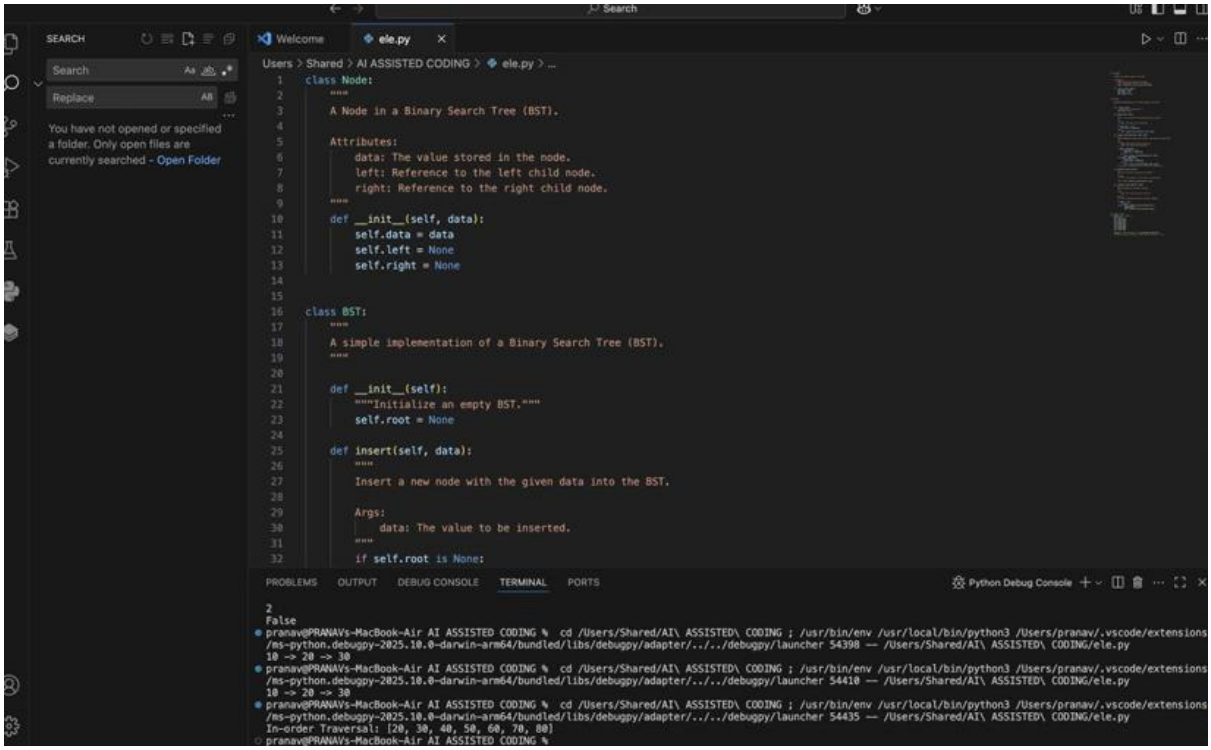


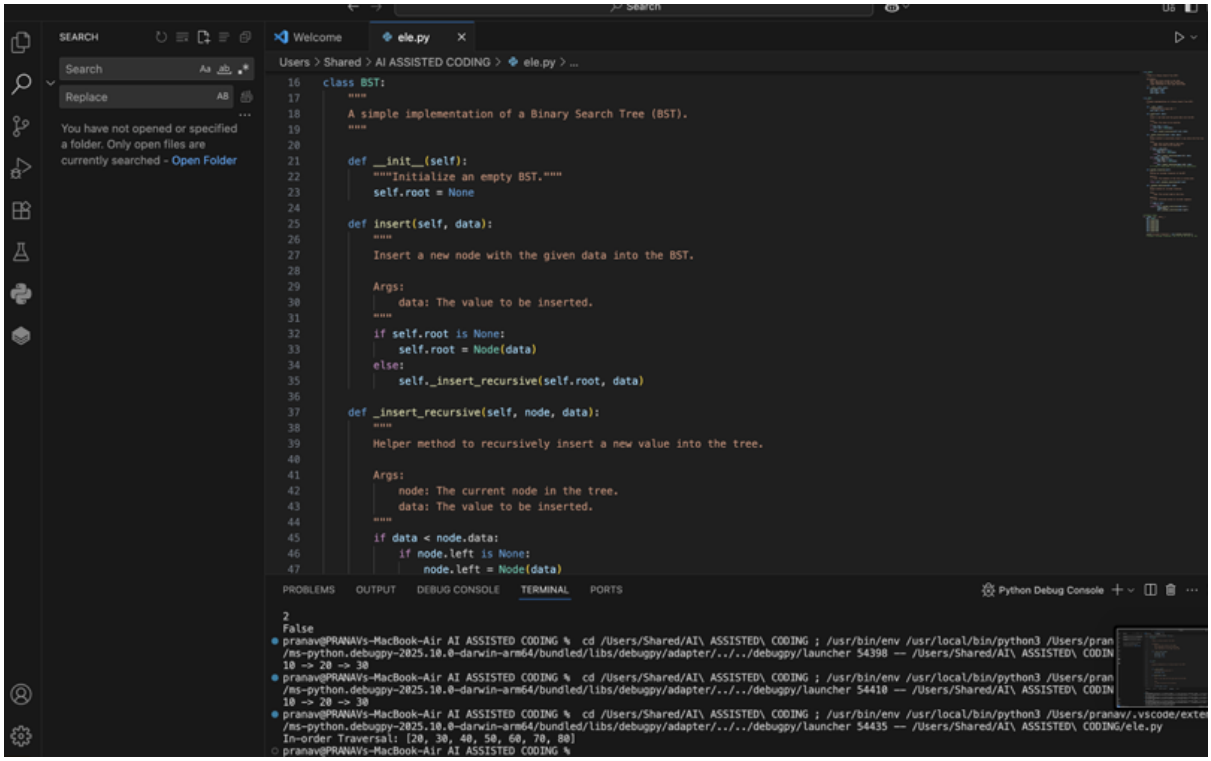
Output:

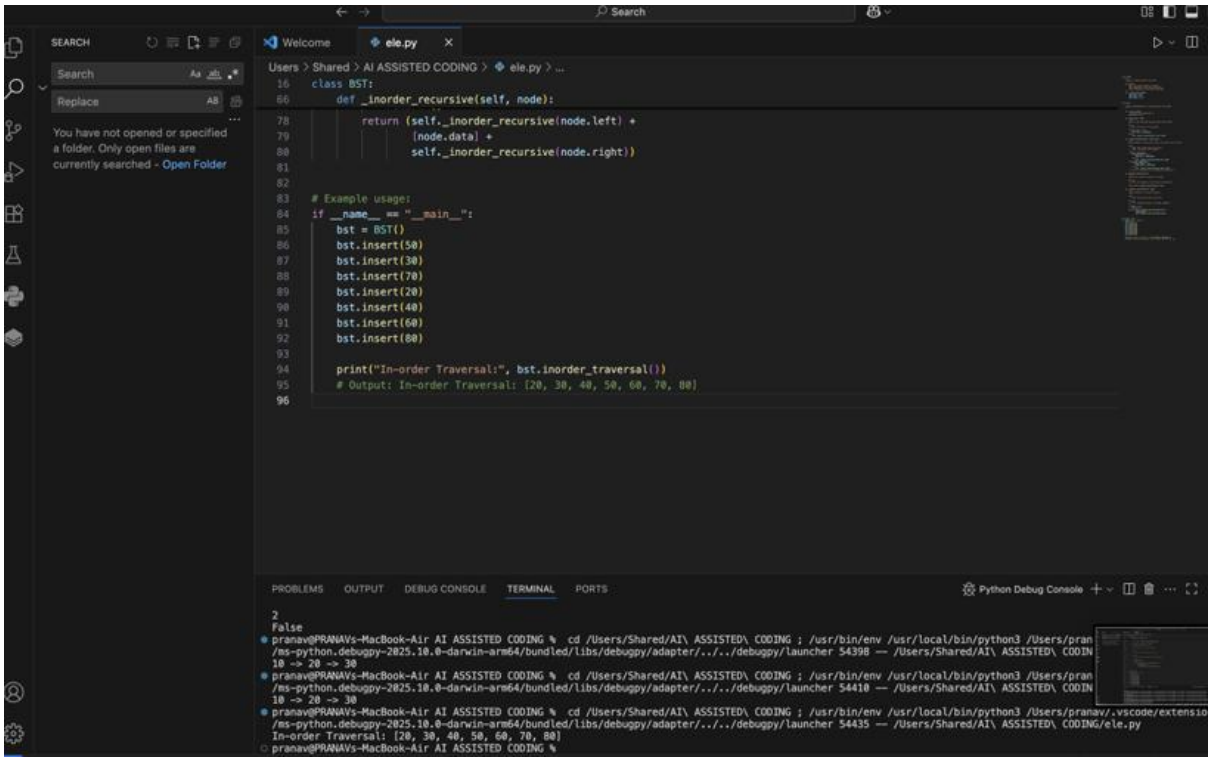
```
/ms-python.debugpy-2025.10.0-darwin-arm64/bundled/libs/debugpy/adapter/../../debugpy/launcher 54362 -- /Users/Shared/AI\
10
10
2
False
pranav@PRANAVs-MacBook-Air AI ASSISTED CODING %  cd /Users/Shared/AI\ ASSISTED\ CODING ; /usr/bin/env /usr/local/bin/pyt
/ms-python.debugpy-2025.10.0-darwin-arm64/bundled/libs/debugpy/adapter/../../debugpy/launcher 54398 -- /Users/Shared/AI\
10 -> 20 -> 30
pranav@PRANAVs-MacBook-Air AI ASSISTED CODING %  cd /Users/Shared/AI\ ASSISTED\ CODING ; /usr/bin/env /usr/local/bin/pyt
/ms-python.debugpy-2025.10.0-darwin-arm64/bundled/libs/debugpy/adapter/../../debugpy/launcher 54410 -- /Users/Shared/AI\
10 -> 20 -> 30
pranav@PRANAVs-MacBook-Air AI ASSISTED CODING %
```

PROMPT : To generate a Singly Linked List with insert and display methods using nodes Task Description #4 – Binary Search Tree (BST) Task: Use AI to create a BST with insert and in-order traversal methods. Sample Input Code: class BST: pass Expected Output: • BST implementation with recursive insert and traversal method

Code:




SEARCH
Search
Replace
You have not opened or specified a folder. Only open files are currently searched - Open Folder
Welcome
ele.py
Users > Shared > AI ASSISTED CODING > ele.py > ...
class BST:
"""
A simple implementation of a Binary Search Tree (BST).
"""
def __init__(self):
"""Initialize an empty BST."""
self.root = None
def insert(self, data):
"""
Insert a new node with the given data into the BST.
Args:
data: The value to be inserted.
"""
if self.root is None:
self.root = Node(data)
else:
self._insert_recursive(self.root, data)
def _insert_recursive(self, node, data):
"""
Helper method to recursively insert a new value into the tree.
Args:
node: The current node in the tree.
data: The value to be inserted.
"""
if data < node.data:
if node.left is None:
node.left = Node(data)
PROBLEMS
OUTPUT
DEBUG CONSOLE
TERMINAL
PORTS
Python Debug Console
2
False
pranav@PRANAVs-MacBook-Air AI ASSISTED CODING % cd /Users/Shared/AI\ ASSISTED\ CODING ; /usr/bin/env /usr/local/bin/python3 /Users/pran
/ms-python.debugpy-2025.10.0-darwin-arm64/bundled/libs/debugpy/adapter/../../debugpy/launcher 54398 -- /Users/Shared/AI\ ASSISTED\ CODIN
10 -> 20 -> 30
pranav@PRANAVs-MacBook-Air AI ASSISTED CODING % cd /Users/Shared/AI\ ASSISTED\ CODING ; /usr/bin/env /usr/local/bin/python3 /Users/pran
/ms-python.debugpy-2025.10.0-darwin-arm64/bundled/libs/debugpy/adapter/../../debugpy/launcher 54410 -- /Users/Shared/AI\ ASSISTED\ CODIN
10 -> 20 -> 30
pranav@PRANAVs-MacBook-Air AI ASSISTED CODING % cd /Users/Shared/AI\ ASSISTED\ CODING ; /usr/bin/env /usr/local/bin/python3 /Users/pranav/.vscode/exte
/ms-python.debugpy-2025.10.0-darwin-arm64/bundled/libs/debugpy/adapter/../../debugpy/launcher 54435 -- /Users/Shared/AI\ ASSISTED\ CODING/ele.py
In-order Traversal: [20, 30, 40, 50, 60, 70, 80]
pranav@PRANAVs-MacBook-Air AI ASSISTED CODING %

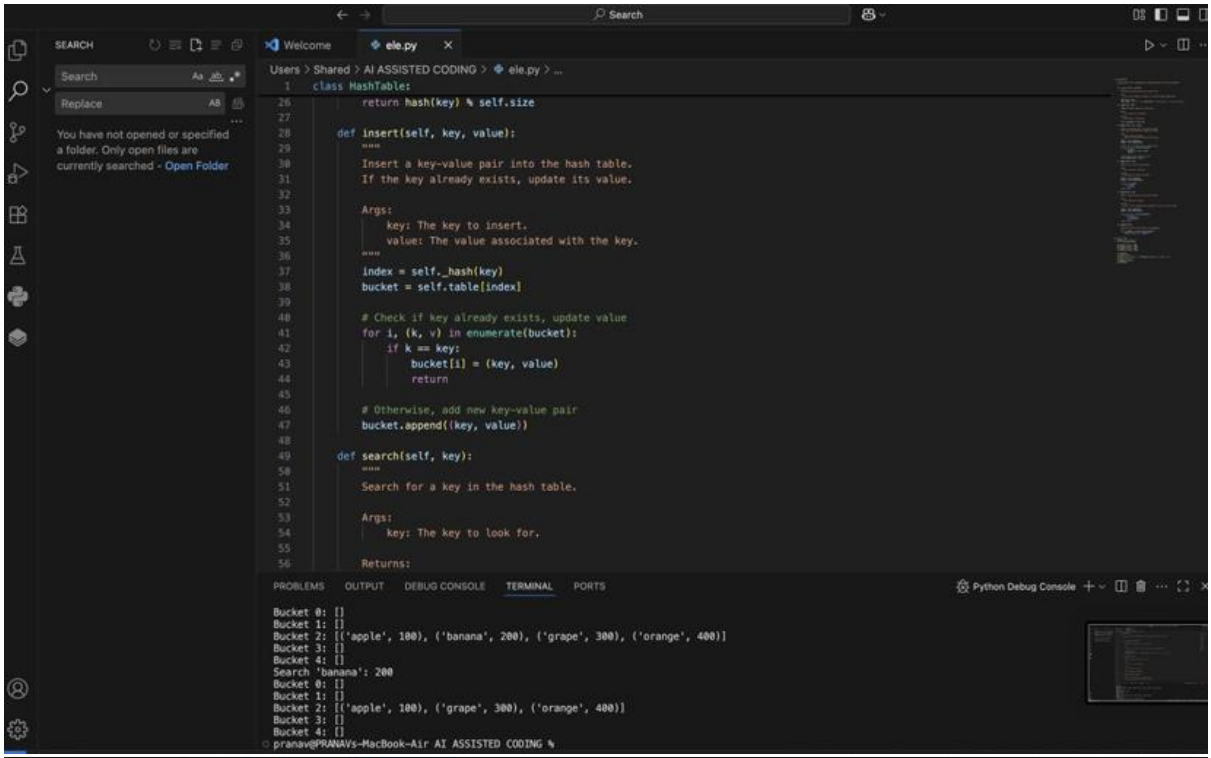

SEARCH
Search
Replace
You have not opened or specified a folder. Only open files are currently searched - Open Folder
Welcome
ele.py
Users > Shared > AI ASSISTED CODING > ele.py > ...
class BST:
def _inorder_recursive(self, node):
return (self._inorder_recursive(node.left) +
[node.data] +
self._inorder_recursive(node.right))
# Example usage:
if __name__ == "__main__":
bst = BST()
bst.insert(50)
bst.insert(30)
bst.insert(70)
bst.insert(20)
bst.insert(40)
bst.insert(60)
bst.insert(80)
print("In-order Traversal:", bst.inorder_traversal())
# Output: In-order Traversal: [20, 30, 40, 50, 60, 70, 80]
PROBLEMS
OUTPUT
DEBUG CONSOLE
TERMINAL
PORTS
Python Debug Console
2
False
pranav@PRANAVs-MacBook-Air AI ASSISTED CODING % cd /Users/Shared/AI\ ASSISTED\ CODING ; /usr/bin/env /usr/local/bin/python3 /Users/pran
/ms-python.debugpy-2025.10.0-darwin-arm64/bundled/libs/debugpy/adapter/../../debugpy/launcher 54398 -- /Users/Shared/AI\ ASSISTED\ CODIN
10 -> 20 -> 30
pranav@PRANAVs-MacBook-Air AI ASSISTED CODING % cd /Users/Shared/AI\ ASSISTED\ CODING ; /usr/bin/env /usr/local/bin/python3 /Users/pran
/ms-python.debugpy-2025.10.0-darwin-arm64/bundled/libs/debugpy/adapter/../../debugpy/launcher 54410 -- /Users/Shared/AI\ ASSISTED\ CODIN
10 -> 20 -> 30
pranav@PRANAVs-MacBook-Air AI ASSISTED CODING % cd /Users/Shared/AI\ ASSISTED\ CODING ; /usr/bin/env /usr/local/bin/python3 /Users/pranav/.vscode/extensio
/ms-python.debugpy-2025.10.0-darwin-arm64/bundled/libs/debugpy/adapter/../../debugpy/launcher 54435 -- /Users/Shared/AI\ ASSISTED\ CODING/ele.py
In-order Traversal: [20, 30, 40, 50, 60, 70, 80]
pranav@PRANAVs-MacBook-Air AI ASSISTED CODING %

Output:


10 -> 20 -> 30
pranav@PRANAVs-MacBook-Air AI ASSISTED CODING % cd /Users/Shared/AI\ ASSISTED\ CODING ; /usr/bin/env /usr/local/bin/python3 /Users/pranav/.vscod
/ms-python.debugpy-2025.10.0-darwin-arm64/bundled/libs/debugpy/adapter/../../debugpy/launcher 54435 -- /Users/Shared/AI\ ASSISTED\ CODING/ele.py
In-order Traversal: [20, 30, 40, 50, 60, 70, 80]
pranav@PRANAVs-MacBook-Air AI ASSISTED CODING %

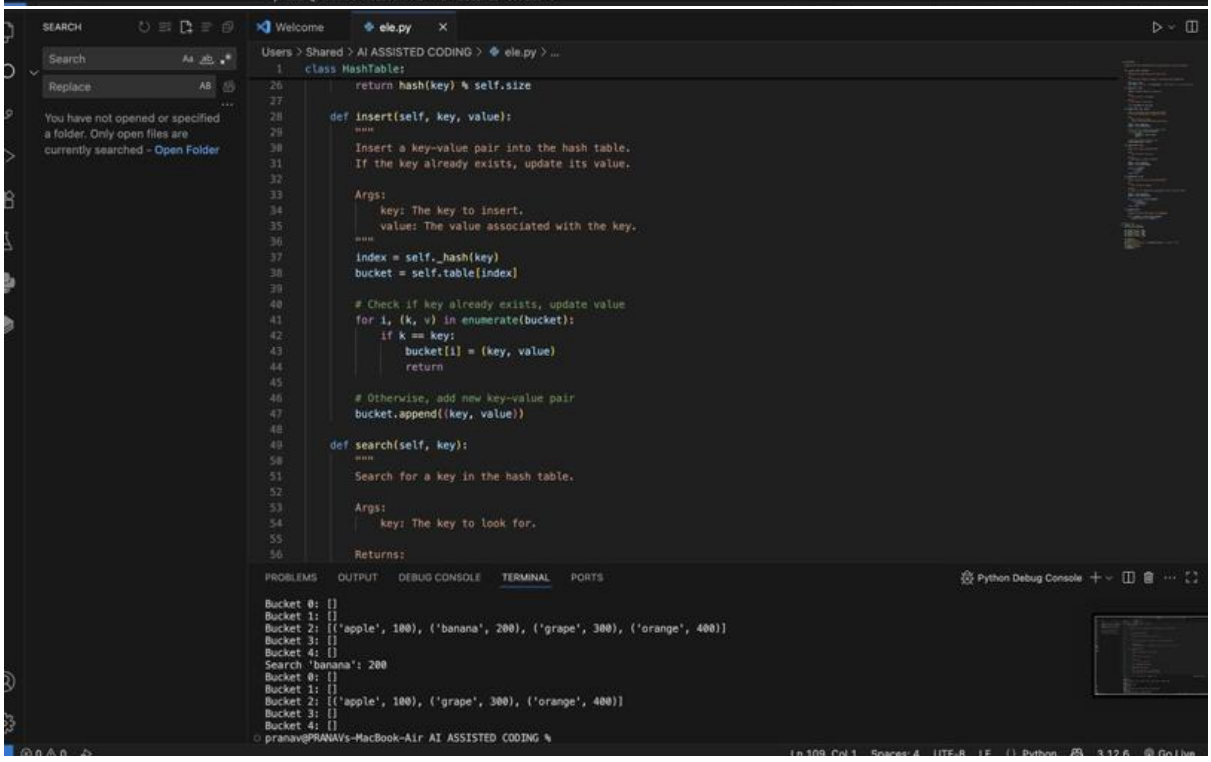
PROMPT : o create a BST with insert and in-order traversal methods. Task Description #5 – Hash Table Task: Use AI to implement a hash table with basic insert, search, and delete methods. Sample Input Code: class HashTable: pass Expected Output: • Collision handling using chaining, with well-commented methods

Code:



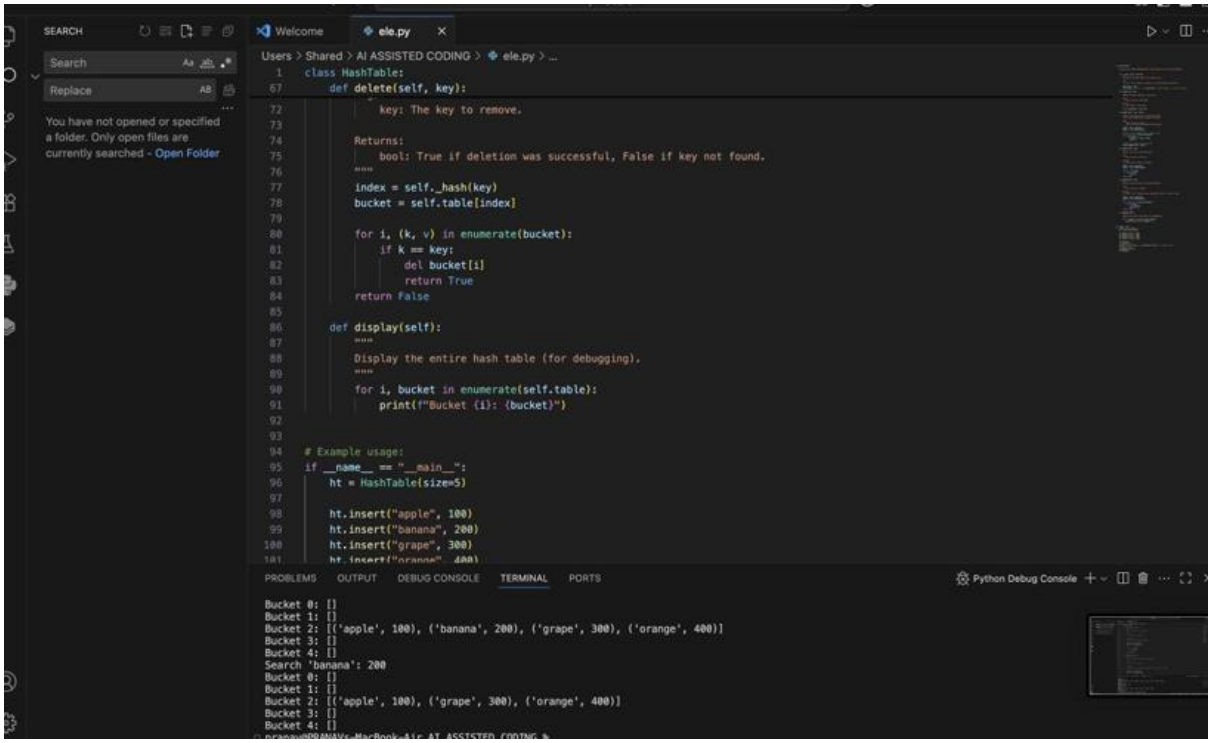
```
class HashTable:
        return hash(key) % self.size

    def insert(self, key, value):
        """
        Insert a key-value pair into the hash table.
        If the key already exists, update its value.

        Args:
            key: The key to insert.
            value: The value associated with the key.
        """
        index = self._hash(key)
        bucket = self.table[index]

        # Check if key already exists, update value
        for i, (k, v) in enumerate(bucket):
            if k == key:
                bucket[i] = (key, value)
                return

        # Otherwise, add new key-value pair
        bucket.append((key, value))

    def search(self, key):
        """
        Search for a key in the hash table.

        Args:
            key: The key to look for.

        Returns:
```

```
Bucket 0: []
Bucket 1: []
Bucket 2: [('apple', 100), ('banana', 200), ('grape', 300), ('orange', 400)]
Bucket 3: []
Bucket 4: []
Search 'banana': 200
Bucket 0: []
Bucket 1: []
Bucket 2: [('apple', 100), ('grape', 300), ('orange', 400)]
Bucket 3: []
Bucket 4: []
pranav@PRANAVs-MacBook-Air AI ASSISTED CODING %
```

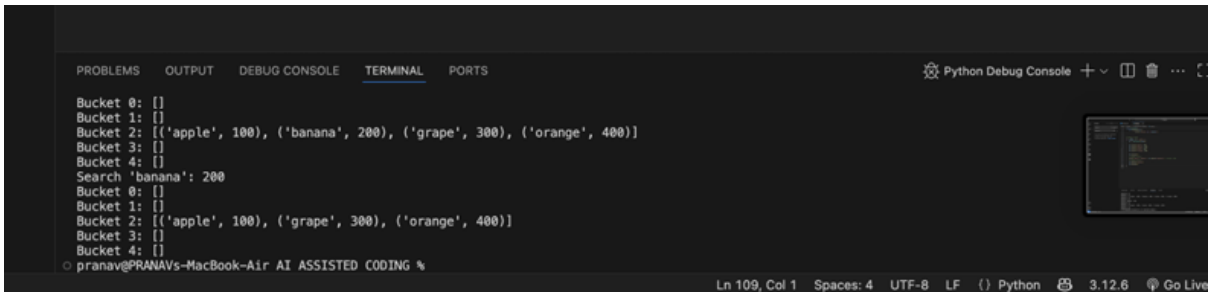



```
class HashTable:
        return hash(key) % self.size

    def insert(self, key, value):
        """
        Insert a key-value pair into the hash table.
        If the key already exists, update its value.

        Args:
            key: The key to insert.
            value: The value associated with the key.
        """
        index = self._hash(key)
        bucket = self.table[index]

        # Check if key already exists, update value
        for i, (k, v) in enumerate(bucket):
            if k == key:
                bucket[i] = (key, value)
                return

        # Otherwise, add new key-value pair
        bucket.append((key, value))

    def search(self, key):
        """
        Search for a key in the hash table.

        Args:
            key: The key to look for.

        Returns:
```

```
Bucket 0: []
Bucket 1: []
Bucket 2: [('apple', 100), ('banana', 200), ('grape', 300), ('orange', 400)]
Bucket 3: []
Bucket 4: []
Search 'banana': 200
Bucket 0: []
Bucket 1: []
Bucket 2: [('apple', 100), ('grape', 300), ('orange', 400)]
Bucket 3: []
Bucket 4: []
pranav@PRANAVs-MacBook-Air AI ASSISTED CODING %
```

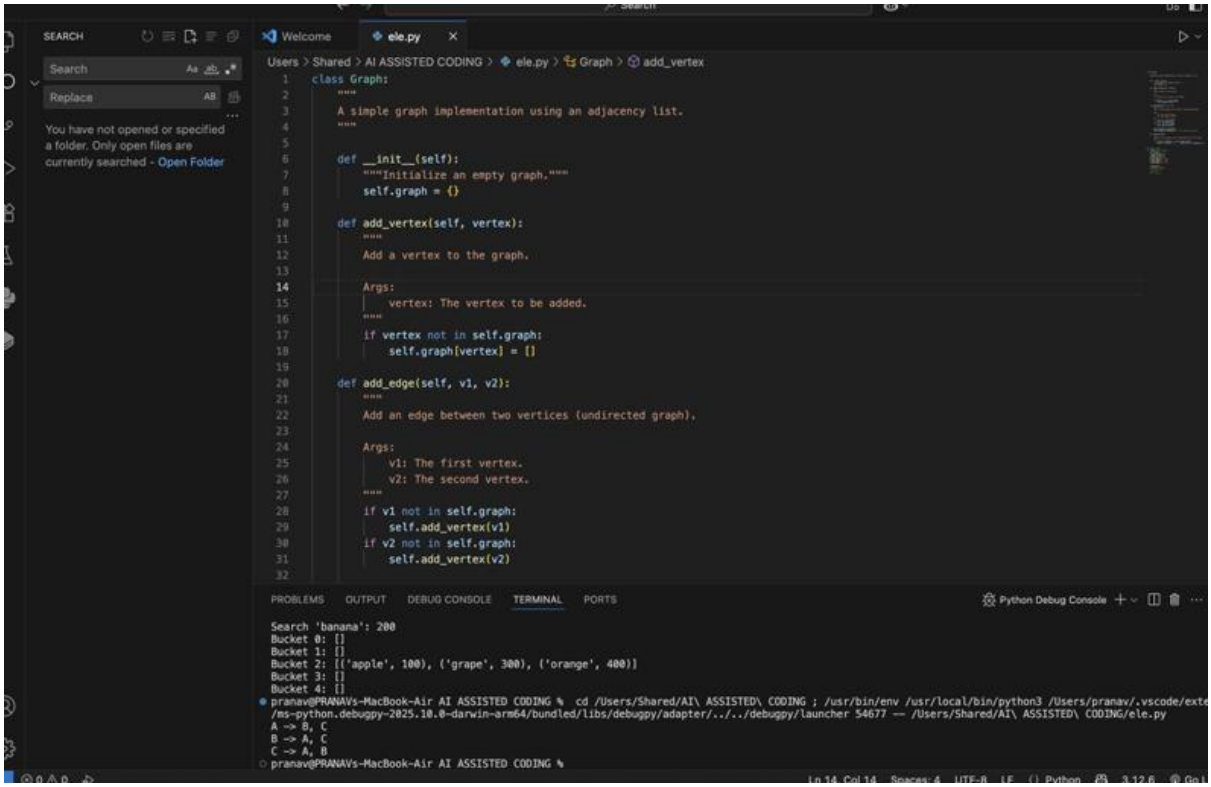


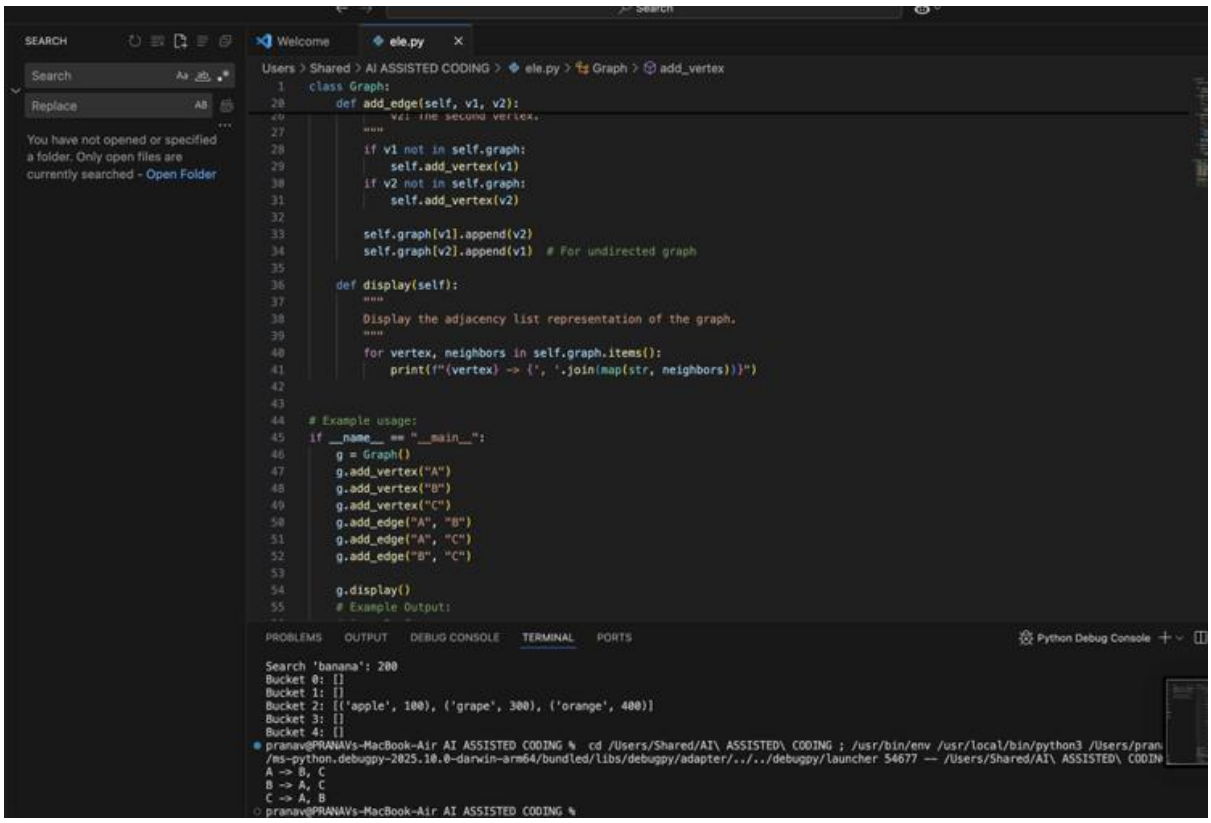
Output:



PROMPT : To implement a hash table with basic insert, search, and delete Methods Task Description #6 – Graph Representation Task: Use AI to implement a graph using an adjacency list. Sample Input Code: class Graph: pass Expected Output: • Graph with methods to add vertices, add edges, and display connections.

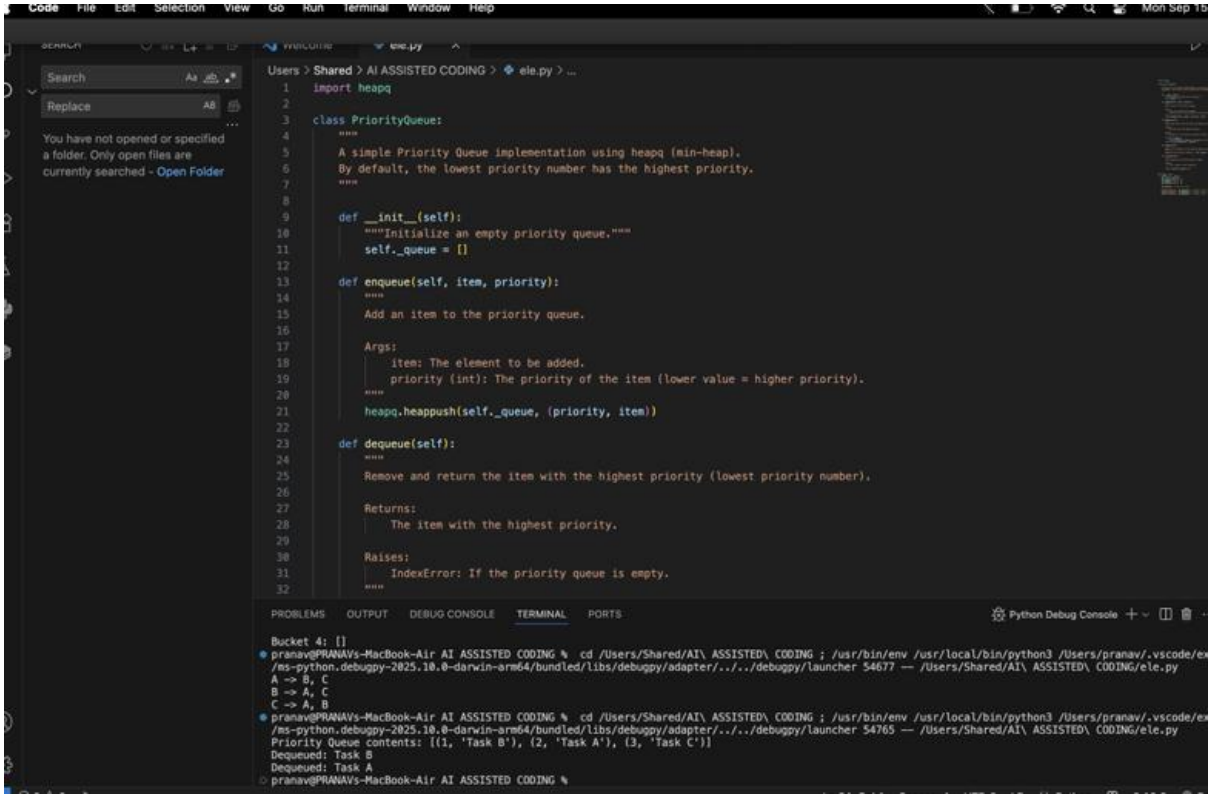
Code:

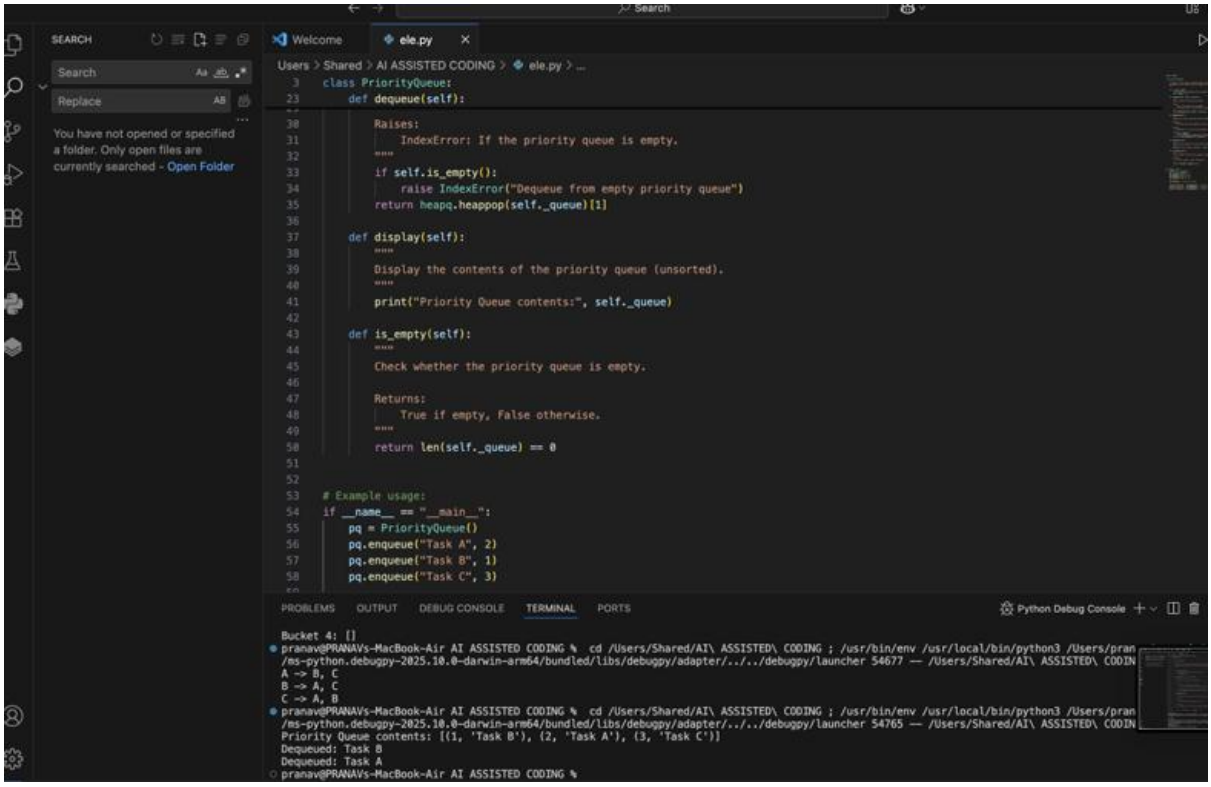

SEARCH
Search
Replace
You have not opened or specified a folder. Only open files are currently searched - Open Folder
Welcome
ele.py
Users > Shared > AI ASSISTED CODING > ele.py > Graph > add_vertex
class Graph:
"""
A simple graph implementation using an adjacency list.
"""
def __init__(self):
"""Initialize an empty graph."""
self.graph = {}
def add_vertex(self, vertex):
"""
Add a vertex to the graph.
Args:
vertex: The vertex to be added.
"""
if vertex not in self.graph:
self.graph[vertex] = []
def add_edge(self, v1, v2):
"""
Add an edge between two vertices (undirected graph).
Args:
v1: The first vertex.
v2: The second vertex.
"""
if v1 not in self.graph:
self.add_vertex(v1)
if v2 not in self.graph:
self.add_vertex(v2)
PROBLEMS
OUTPUT
DEBUG CONSOLE
TERMINAL
PORTS
Python Debug Console
Search 'banana': 200
Bucket 0: []
Bucket 1: []
Bucket 2: [('apple', 100), ('grape', 300), ('orange', 400)]
Bucket 3: []
Bucket 4: []
pranav@PRANAVs-MacBook-Air AI ASSISTED CODING % cd /Users/Shared/AI\ ASSISTED\ CODING ; /usr/bin/env /usr/local/bin/python3 /Users/pranav/.vscode/exte
/ms-python.debugpy-2025.10.0-darwin-arm64/bundled/libs/debugpy/adapter/../../debugpy/launcher 54677 -- /Users/Shared/AI\ ASSISTED\ CODING/ele.py
A -> B, C
B -> A, C
C -> A, B
pranav@PRANAVs-MacBook-Air AI ASSISTED CODING %


SEARCH
Search
Replace
You have not opened or specified a folder. Only open files are currently searched - Open Folder
Welcome
ele.py
Users > Shared > AI ASSISTED CODING > ele.py > Graph > add_vertex
class Graph:
def add_edge(self, v1, v2):
"""
if v1 not in self.graph:
self.add_vertex(v1)
if v2 not in self.graph:
self.add_vertex(v2)
self.graph[v1].append(v2)
self.graph[v2].append(v1) # For undirected graph
def display(self):
"""
Display the adjacency list representation of the graph.
"""
for vertex, neighbors in self.graph.items():
print(f"{vertex} -> {', '.join(map(str, neighbors))}")
# Example usage:
if __name__ == "__main__":
g = Graph()
g.add_vertex("A")
g.add_vertex("B")
g.add_vertex("C")
g.add_edge("A", "B")
g.add_edge("A", "C")
g.add_edge("B", "C")
g.display()
# Example Output:
PROBLEMS
OUTPUT
DEBUG CONSOLE
TERMINAL
PORTS
Python Debug Console
Search 'banana': 200
Bucket 0: []
Bucket 1: []
Bucket 2: [('apple', 100), ('grape', 300), ('orange', 400)]
Bucket 3: []
Bucket 4: []
pranav@PRANAVs-MacBook-Air AI ASSISTED CODING % cd /Users/Shared/AI\ ASSISTED\ CODING ; /usr/bin/env /usr/local/bin/python3 /Users/pran
/ms-python.debugpy-2025.10.0-darwin-arm64/bundled/libs/debugpy/adapter/../../debugpy/launcher 54677 -- /Users/Shared/AI\ ASSISTED\ CODIN
A -> B, C
B -> A, C
C -> A, B
pranav@PRANAVs-MacBook-Air AI ASSISTED CODING %

Task Description #7 – Priority Queue Task: Use AI to implement a priority queue using Python's heapq module. Sample Input Code: class PriorityQueue: pass Expected Output: • Implementation with enqueue (priority), dequeue (highest priority), and display methods

Code:

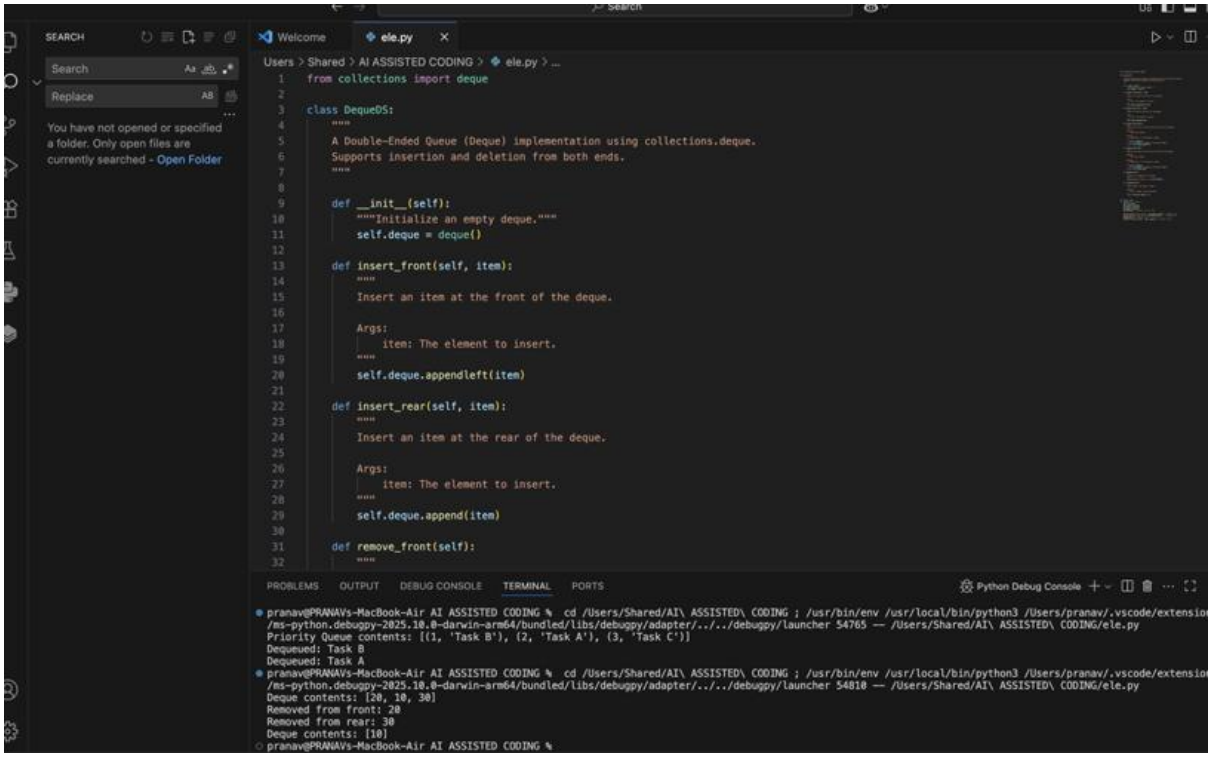


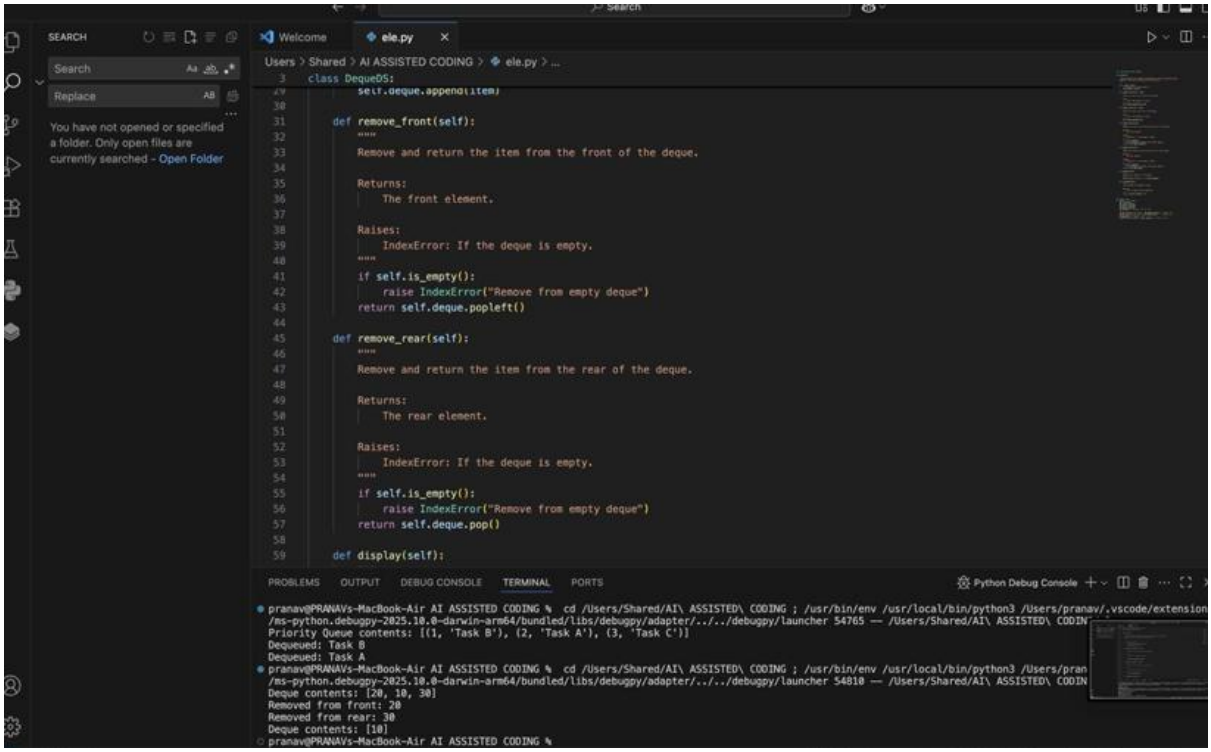


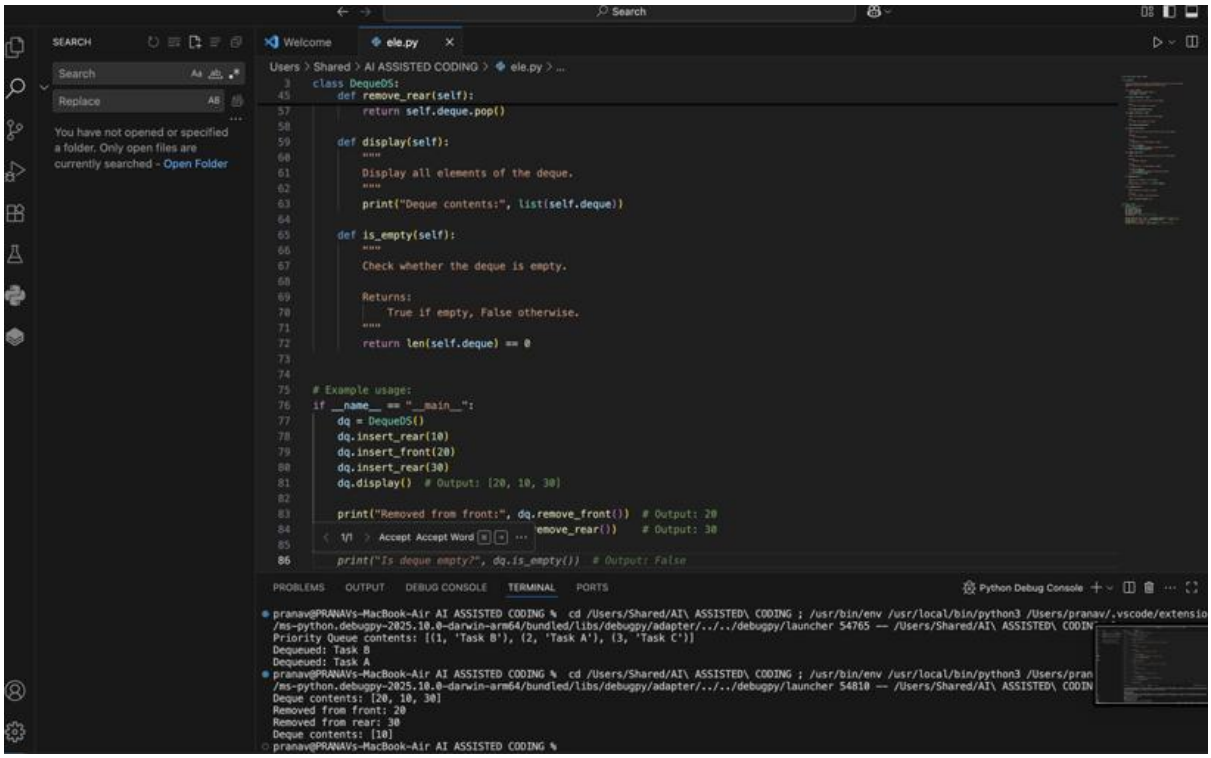
OUTPUT :

PROMPT : To implement a priority queue using Python’s heapq module. Task Description #8 – Deque Task: Use AI to implement a double-ended queue using collections.deque. Sample Input Code: class DequeDS: pass Expected Output: • Insert and remove from both ends with docstrings

Code:


SEARCH
Search
Replace
You have not opened or specified a folder. Only open files are currently searched - Open Folder
Welcome
ele.py
Users > Shared > AI ASSISTED CODING > ele.py > ...
from collections import deque
class DequeDS:
"""
A Double-Ended Queue (Deque) implementation using collections.deque.
Supports insertion and deletion from both ends.
"""
def __init__(self):
"""Initialize an empty deque."""
self.deque = deque()
def insert_front(self, item):
"""
Insert an item at the front of the deque.
Args:
item: The element to insert.
"""
self.deque.appendleft(item)
def insert_rear(self, item):
"""
Insert an item at the rear of the deque.
Args:
item: The element to insert.
"""
self.deque.append(item)
def remove_front(self):
"""
PROBLEMS
OUTPUT
DEBUG CONSOLE
TERMINAL
PORTS
Python Debug Console
pranav@PRANAVs-MacBook-Air AI ASSISTED CODING % cd /Users/Shared/AI\ ASSISTED\ CODING ; /usr/bin/env /usr/local/bin/python3 /Users/pranav/.vscode/extensio
/ms-python.debugpy-2025.10.0-darwin-arm64/bundled/libs/debugpy/adapter/../../debugpy/launcher 54765 -- /Users/Shared/AI\ ASSISTED\ CODING/ele.py
Priority Queue contents: [(1, 'Task B'), (2, 'Task A'), (3, 'Task C')]
Dequeued: Task B
Dequeued: Task A
pranav@PRANAVs-MacBook-Air AI ASSISTED CODING % cd /Users/Shared/AI\ ASSISTED\ CODING ; /usr/bin/env /usr/local/bin/python3 /Users/pranav/.vscode/extensio
/ms-python.debugpy-2025.10.0-darwin-arm64/bundled/libs/debugpy/adapter/../../debugpy/launcher 54810 -- /Users/Shared/AI\ ASSISTED\ CODING/ele.py
Deque contents: [20, 10, 30]
Removed from front: 20
Removed from rear: 30
Deque contents: [10]
pranav@PRANAVs-MacBook-Air AI ASSISTED CODING %

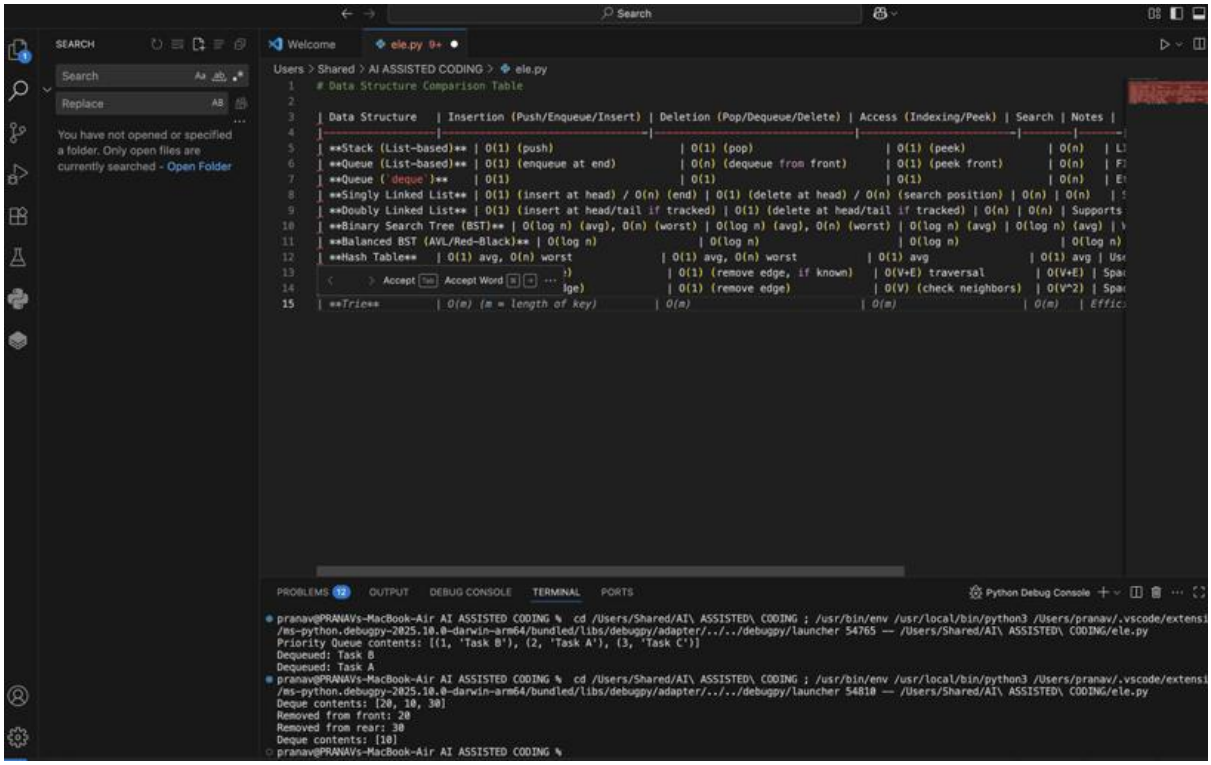

SEARCH
Search
Replace
You have not opened or specified a folder. Only open files are currently searched - Open Folder
Welcome
ele.py
Users > Shared > AI ASSISTED CODING > ele.py > ...
class DequeDS:
self.deque.append(item)
def remove_front(self):
"""
Remove and return the item from the front of the deque.
Returns:
The front element.
Raises:
IndexError: If the deque is empty.
"""
if self.is_empty():
raise IndexError("Remove from empty deque")
return self.deque.popleft()
def remove_rear(self):
"""
Remove and return the item from the rear of the deque.
Returns:
The rear element.
Raises:
IndexError: If the deque is empty.
"""
if self.is_empty():
raise IndexError("Remove from empty deque")
return self.deque.pop()
def display(self):
PROBLEMS
OUTPUT
DEBUG CONSOLE
TERMINAL
PORTS
Python Debug Console
pranav@PRANAVs-MacBook-Air AI ASSISTED CODING % cd /Users/Shared/AI\ ASSISTED\ CODING ; /usr/bin/env /usr/local/bin/python3 /Users/pranav/.vscode/extension
/ms-python.debugpy-2025.10.0-darwin-arm64/bundled/libs/debugpy/adapter/../../debugpy/launcher 54765 -- /Users/Shared/AI\ ASSISTED\ CODIN
Priority Queue contents: [(1, 'Task B'), (2, 'Task A'), (3, 'Task C')]
Dequeued: Task B
Dequeued: Task A
pranav@PRANAVs-MacBook-Air AI ASSISTED CODING % cd /Users/Shared/AI\ ASSISTED\ CODING ; /usr/bin/env /usr/local/bin/python3 /Users/pran
/ms-python.debugpy-2025.10.0-darwin-arm64/bundled/libs/debugpy/adapter/../../debugpy/launcher 54810 -- /Users/Shared/AI\ ASSISTED\ CODIN
Deque contents: [20, 10, 30]
Removed from front: 20
Removed from rear: 30
Deque contents: [10]
pranav@PRANAVs-MacBook-Air AI ASSISTED CODING %

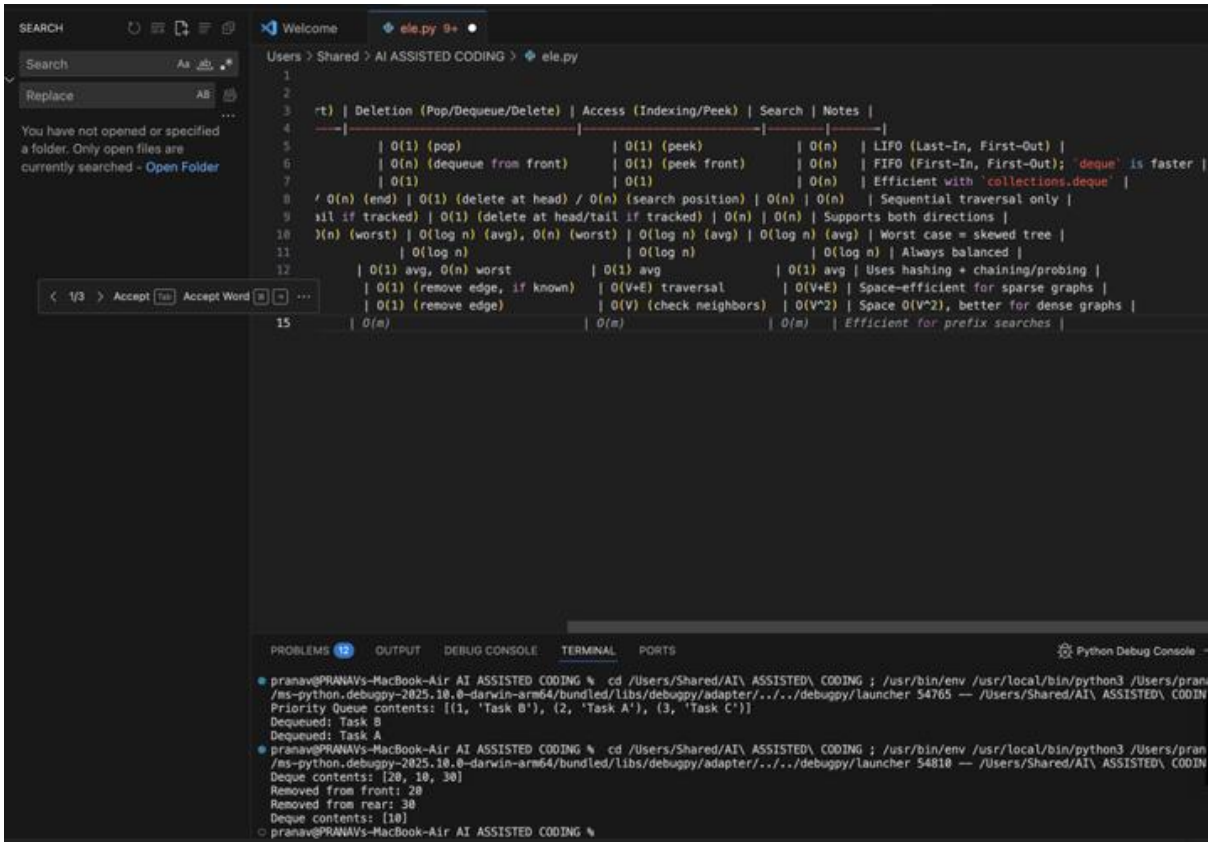

SEARCH
Search
Replace
You have not opened or specified a folder. Only open files are currently searched - Open Folder
Welcome
ele.py
Users > Shared > AI ASSISTED CODING > ele.py > ...
class DequeDS:
def remove_rear(self):
return self.deque.pop()
def display(self):
"""
Display all elements of the deque.
"""
print("Deque contents:", list(self.deque))
def is_empty(self):
"""
Check whether the deque is empty.
Returns:
True if empty, False otherwise.
"""
return len(self.deque) == 0
# Example usage:
if __name__ == "__main__":
dq = DequeDS()
dq.insert_rear(10)
dq.insert_front(20)
dq.insert_rear(30)
dq.display() # Output: [20, 10, 30]
print("Removed from front:", dq.remove_front()) # Output: 20
emove_rear()) # Output: 30
1/1
Accept
Accept Word
print("Is deque empty?", dq.is_empty()) # Output: False
PROBLEMS
OUTPUT
DEBUG CONSOLE
TERMINAL
PORTS
Python Debug Console
pranav@PRANAVs-MacBook-Air AI ASSISTED CODING % cd /Users/Shared/AI\ ASSISTED\ CODING ; /usr/bin/env /usr/local/bin/python3 /Users/pranav/.vscode/extensio
/ms-python.debugpy-2025.10.0-darwin-arm64/bundled/libs/debugpy/adapter/../../debugpy/launcher 54765 -- /Users/Shared/AI\ ASSISTED\ CODIN
Priority Queue contents: [(1, 'Task B'), (2, 'Task A'), (3, 'Task C')]
Dequeued: Task B
Dequeued: Task A
pranav@PRANAVs-MacBook-Air AI ASSISTED CODING % cd /Users/Shared/AI\ ASSISTED\ CODING ; /usr/bin/env /usr/local/bin/python3 /Users/pran
/ms-python.debugpy-2025.10.0-darwin-arm64/bundled/libs/debugpy/adapter/../../debugpy/launcher 54810 -- /Users/Shared/AI\ ASSISTED\ CODIN
Deque contents: [20, 10, 30]
Removed from front: 20
Removed from rear: 30
Deque contents: [10]
pranav@PRANAVs-MacBook-Air AI ASSISTED CODING %

Output:

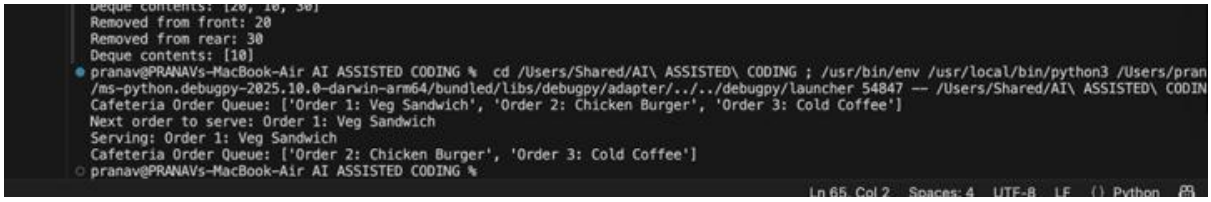
PROMPT : To implement a double-ended queue using collections.deque Task Description #9 – AI-Generated Data Structure Comparisons Task: Use AI to generate a comparison table of different data structures (stack, queue, linked list, etc.) including time complexities. Sample Input Code: # No code, prompt AI for a data structure comparison table Expected Output: • A markdown table with structure names, operations, and complexities

Code:





Output:



PROMPT : Your college wants to develop a Campus Resource Management System that handles: 1. Student Attendance Tracking – Daily log of students entering/exiting the campus. 2. Event Registration System – Manage participants in events with quick search and removal. 3. Library Book Borrowing – Keep track of available books and their due dates. 4. Bus Scheduling System – Maintain bus routes and stop connections. 5. Cafeteria Order Queue – Serve students in the order they arrive. Student Task: • For each feature, select the most appropriate data structure from the list below: o Stack o Queue o Priority Queue o Linked List o Binary Search

Tree (BST) o Graph o Hash Table o Deque • Justify your choice in 2–3 sentences per feature. • Implement one selected feature as a working Python program with AI- assisted code generation.