NAME : P SAI VENKAT ROLL NO :2403A510G0

SUBJECT: AI ASSISTED CODING ASSIGNMENT : 11.1

Task Description #1 – Stack Implementation

Task: Use AI to generate a Stack class with push, pop, peek, and is_empty methods.

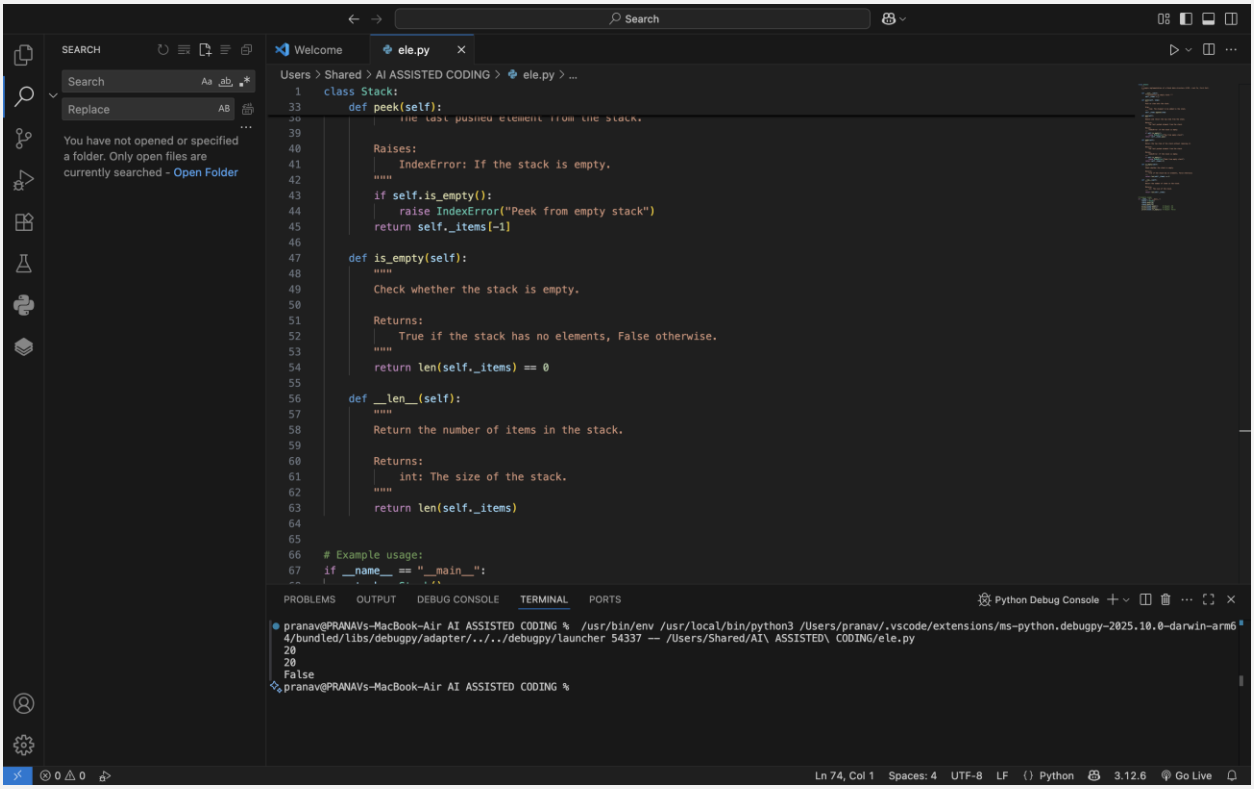
Sample Input Code:

class Stack:

pass

Expected Output:

• A functional stack implementation with all required methods and docstrings.




OUTPUT :

```
    def __len__(self):
        """
        Return the number of items in the stack.

        Returns:
            int: The size of the stack.
        """
        return len(self._items)


# Example usage:
if __name__ == "__main__":
```

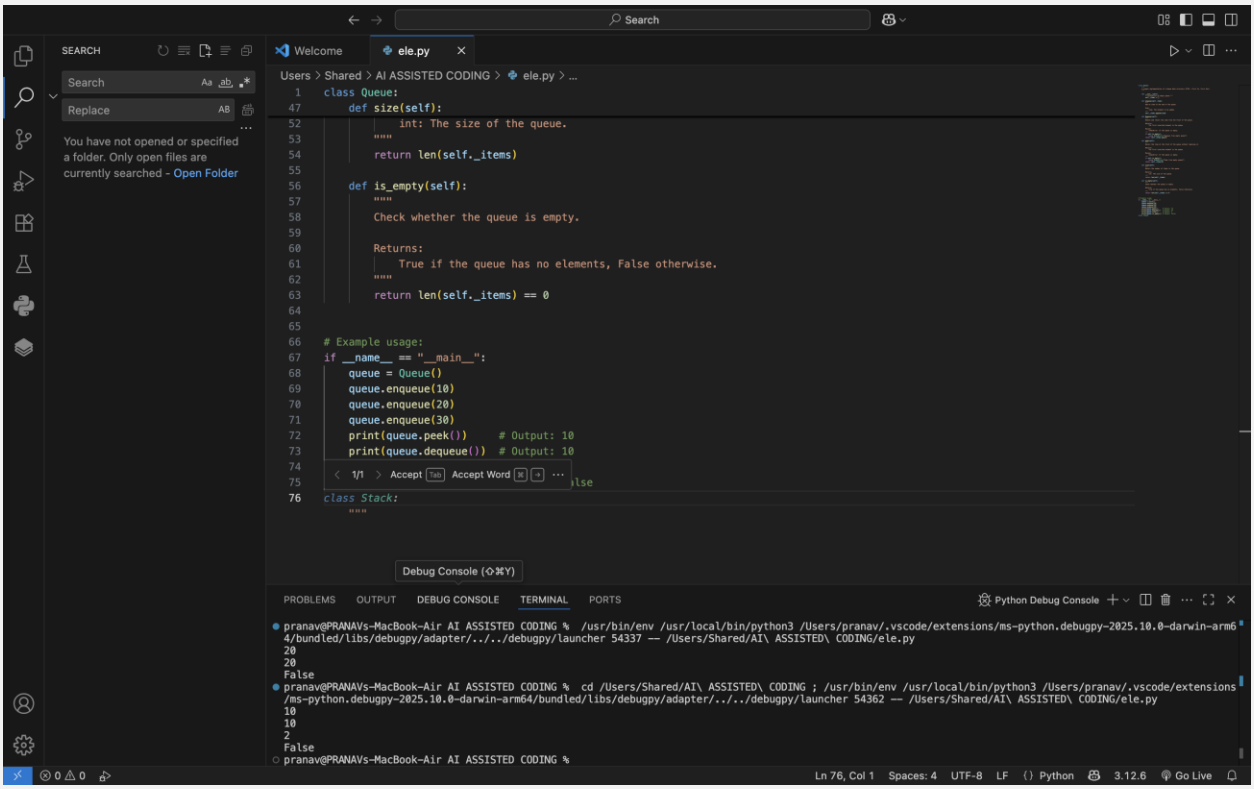
```
PROBLEMS    OUTPUT    DEBUG CONSOLE    TERMINAL    PORTS                    Python Debug Console

pranav@PRANAVs-MacBook-Air AI ASSISTED CODING %  /usr/bin/env /usr/local/bin/python3 /Users/pranav/.vscode/extensions/ms-python.debugpy-2025.10.0-darwin-arm64/bundled/libs/debugpy/adapter/../../debugpy/launcher 54337 -- /Users/Shared/AI\ ASSISTED\ CODING/ele.py
20
20
False
pranav@PRANAVs-MacBook-Air AI ASSISTED CODING %
```

Ln 74, Col 1  Spaces: 4  UTF-8  LF  {} Python  3.12.6  Go Live

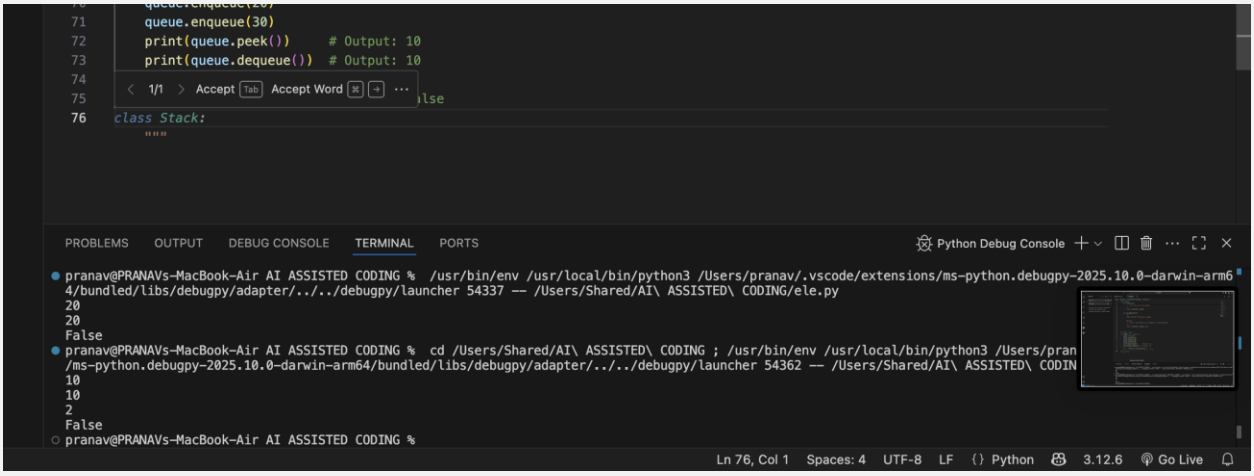
PROMPT :

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




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:

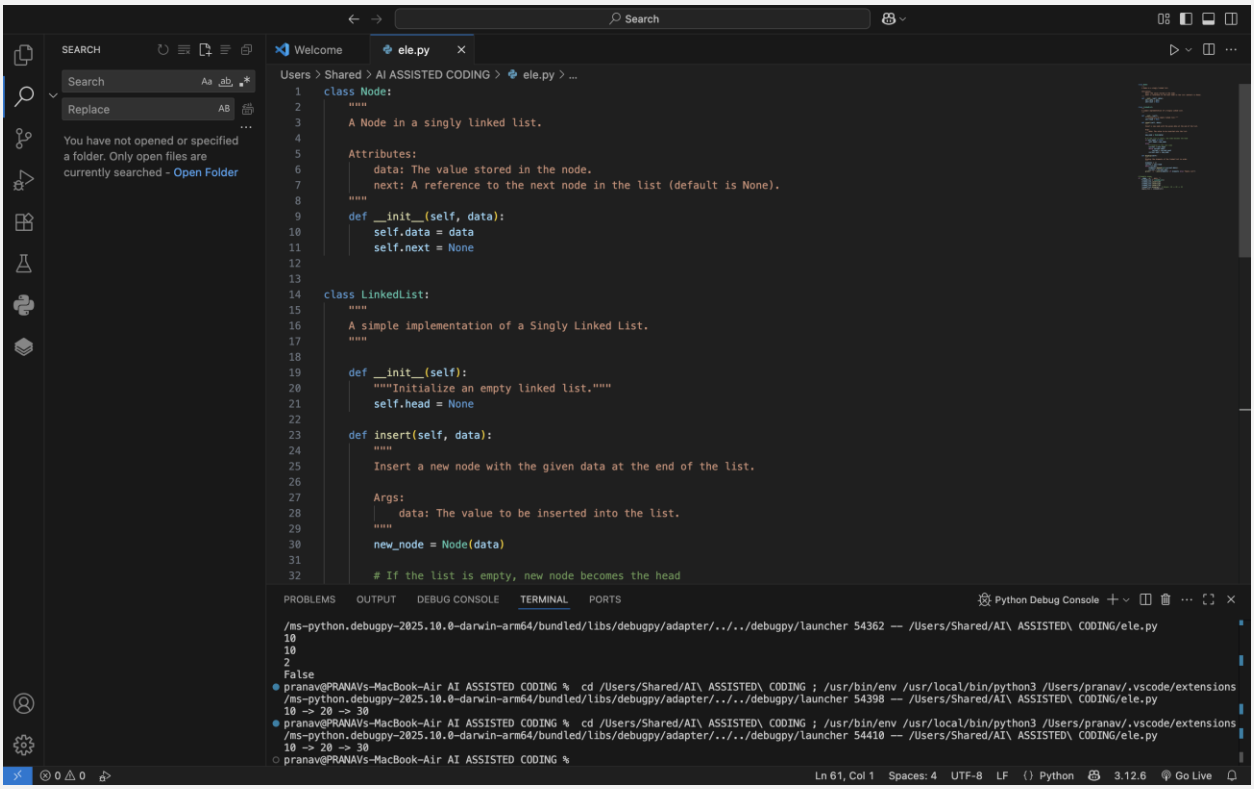
pass

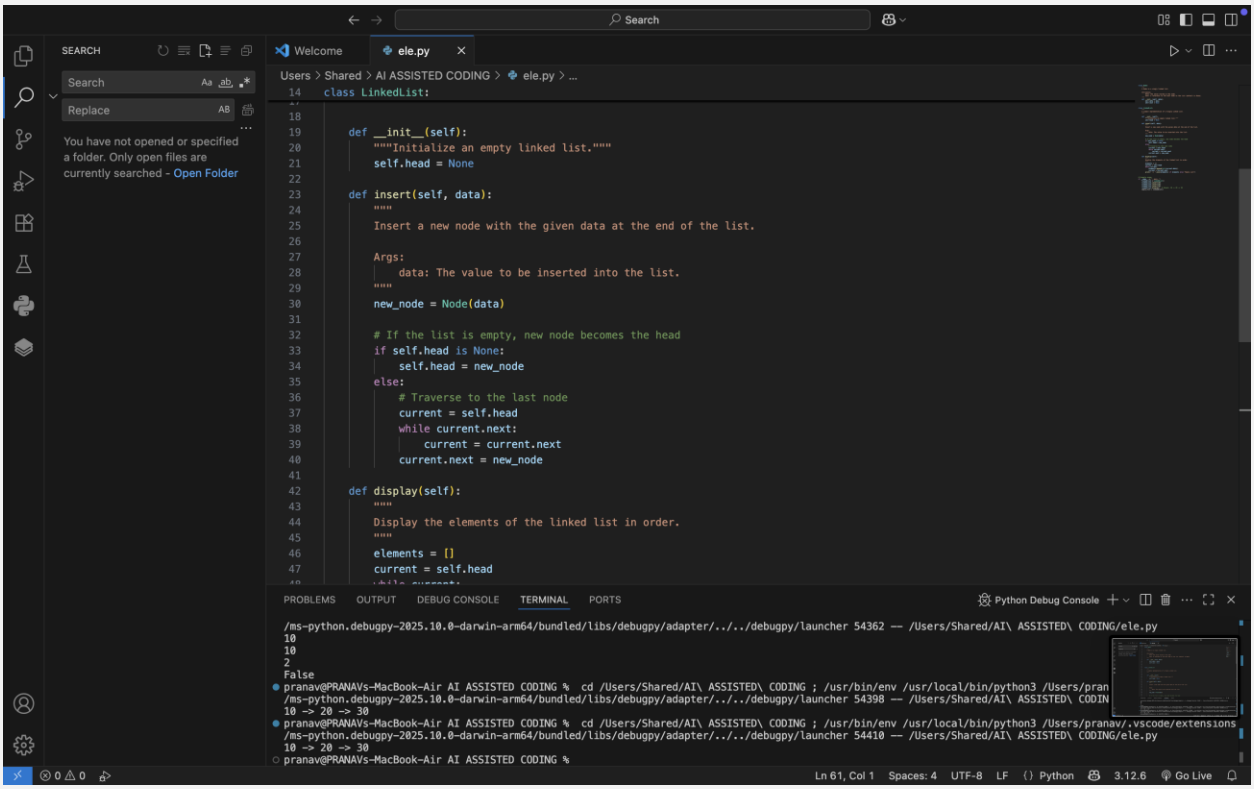
class LinkedList:

pass

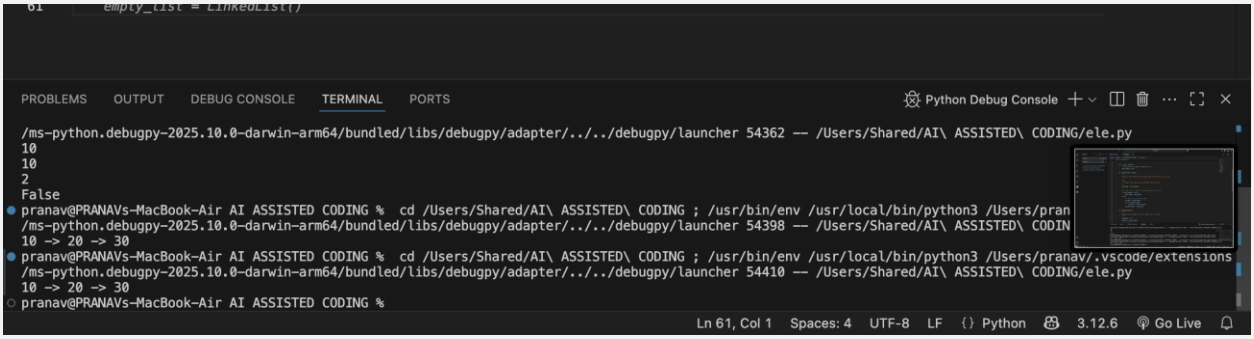
Expected Output:

• A working linked list implementation with clear method documentation.






OUTPUT :




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.

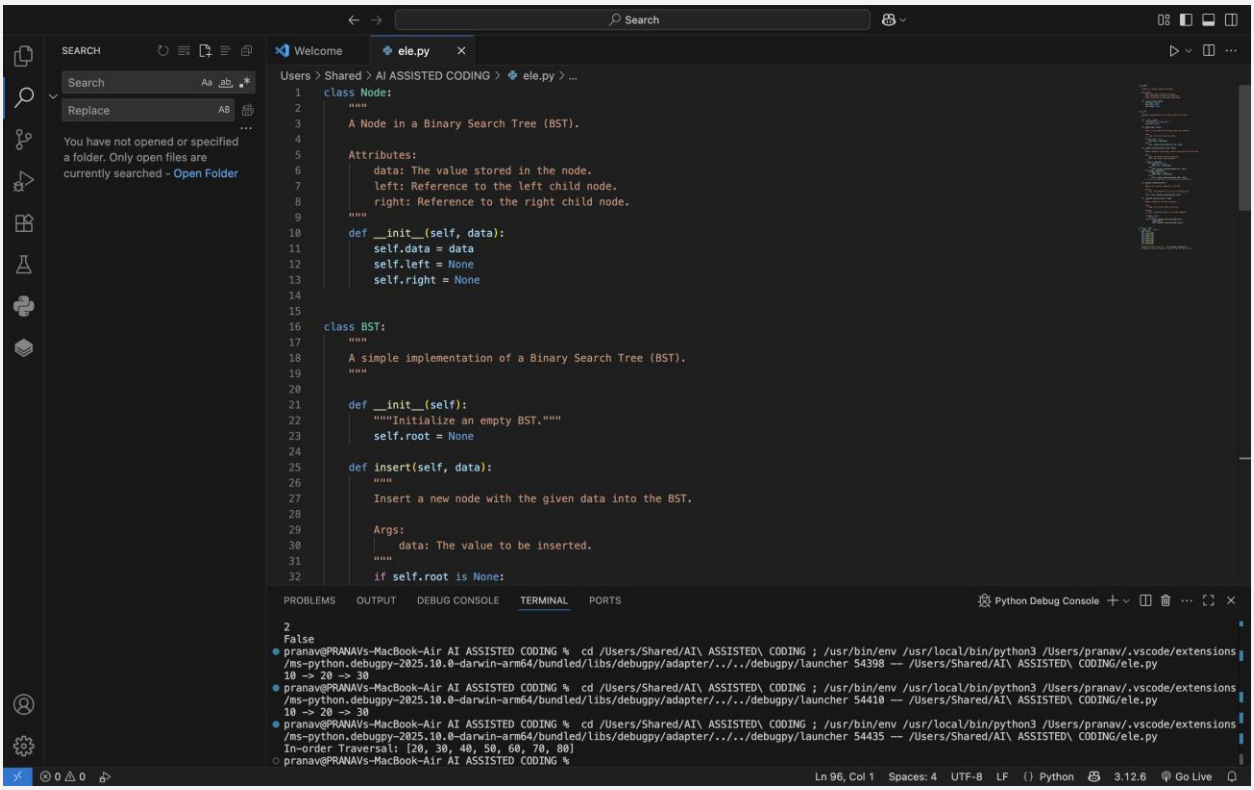
Sample Input Code:

class BST:

pass

Expected Output:

• BST implementation with recursive insert and traversal method



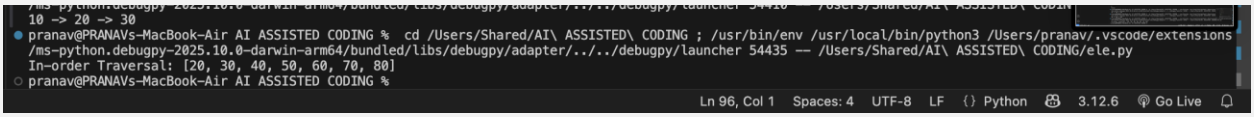
```python
class BST:
    """
    A simple implementation of a Binary Search Tree (BST).
    """

    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)
```

```
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/extensions
/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 %
```

```python
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]
```

```
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/extensions
/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 :




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.

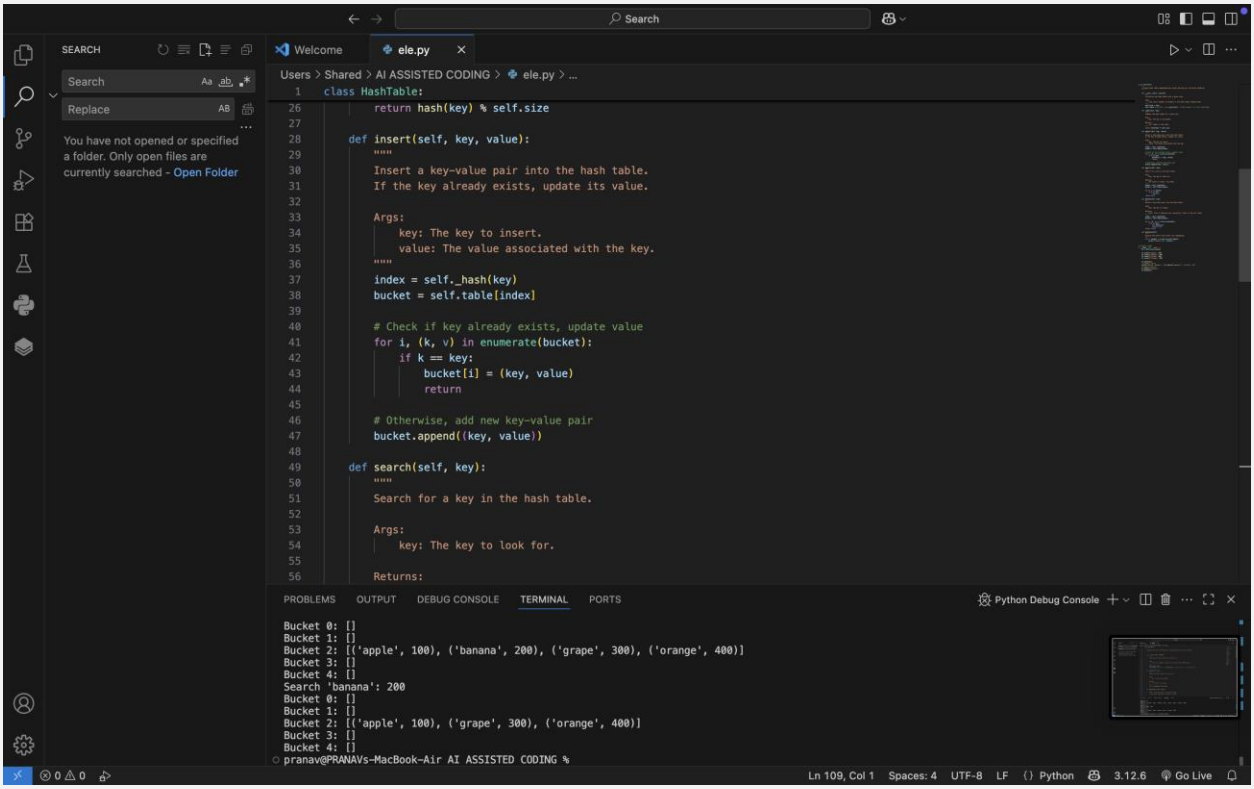
Sample Input Code:

class HashTable:

pass

Expected Output:

• Collision handling using chaining, with well-commented methods



Users > Shared > AI ASSISTED CODING > ele.py > ...

```python
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:
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

```
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 %
```

Ln 109, Col 1 Spaces: 4 UTF-8 LF Python 3.12.6 Go Live

Users > Shared > AI ASSISTED CODING > ele.py > ...

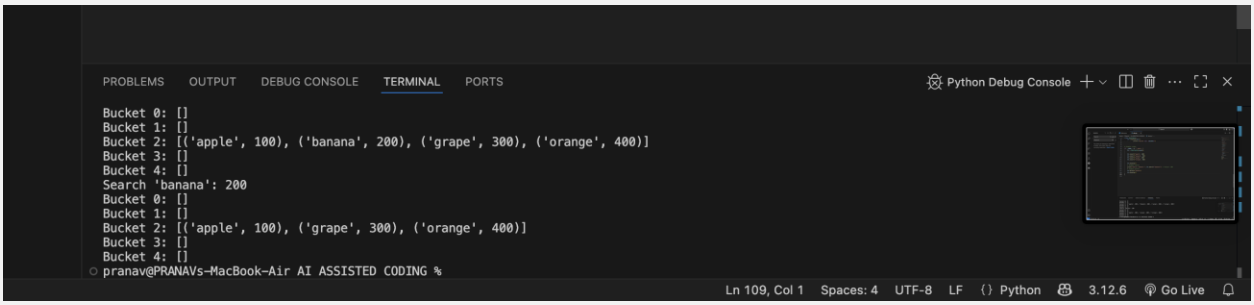
```python
class HashTable:
    def delete(self, key):
            key: The key to remove.

        Returns:
            bool: True if deletion was successful, False if key not found.
        """
        index = self._hash(key)
        bucket = self.table[index]

        for i, (k, v) in enumerate(bucket):
            if k == key:
                del bucket[i]
                return True
        return False

    def display(self):
        """
        Display the entire hash table (for debugging).
        """
        for i, bucket in enumerate(self.table):
            print(f"Bucket {i}: {bucket}")


# Example usage:
if __name__ == "__main__":
    ht = HashTable(size=5)

    ht.insert("apple", 100)
    ht.insert("banana", 200)
    ht.insert("grape", 300)
    ht.insert("orange", 400)
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

```
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 %
```

Ln 109, Col 1 Spaces: 4 UTF-8 LF Python 3.12.6 Go Live

OUTPUT :

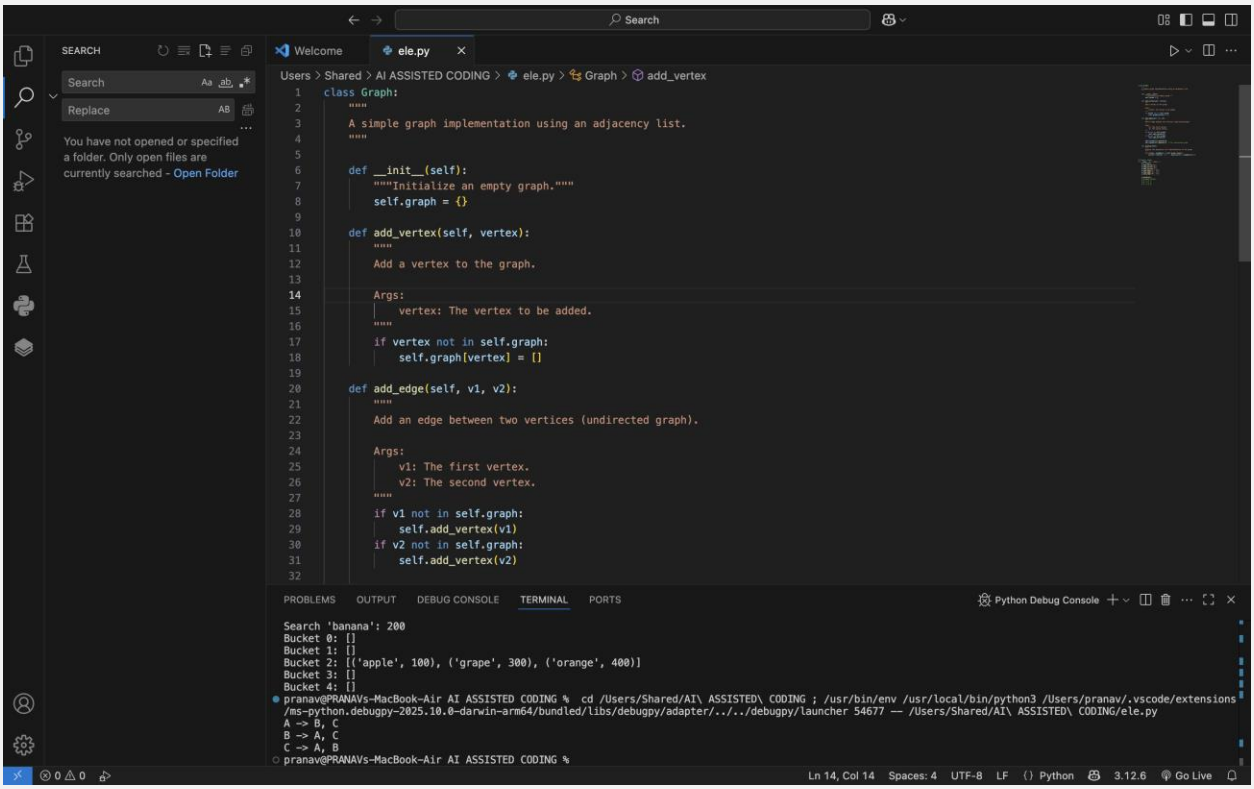


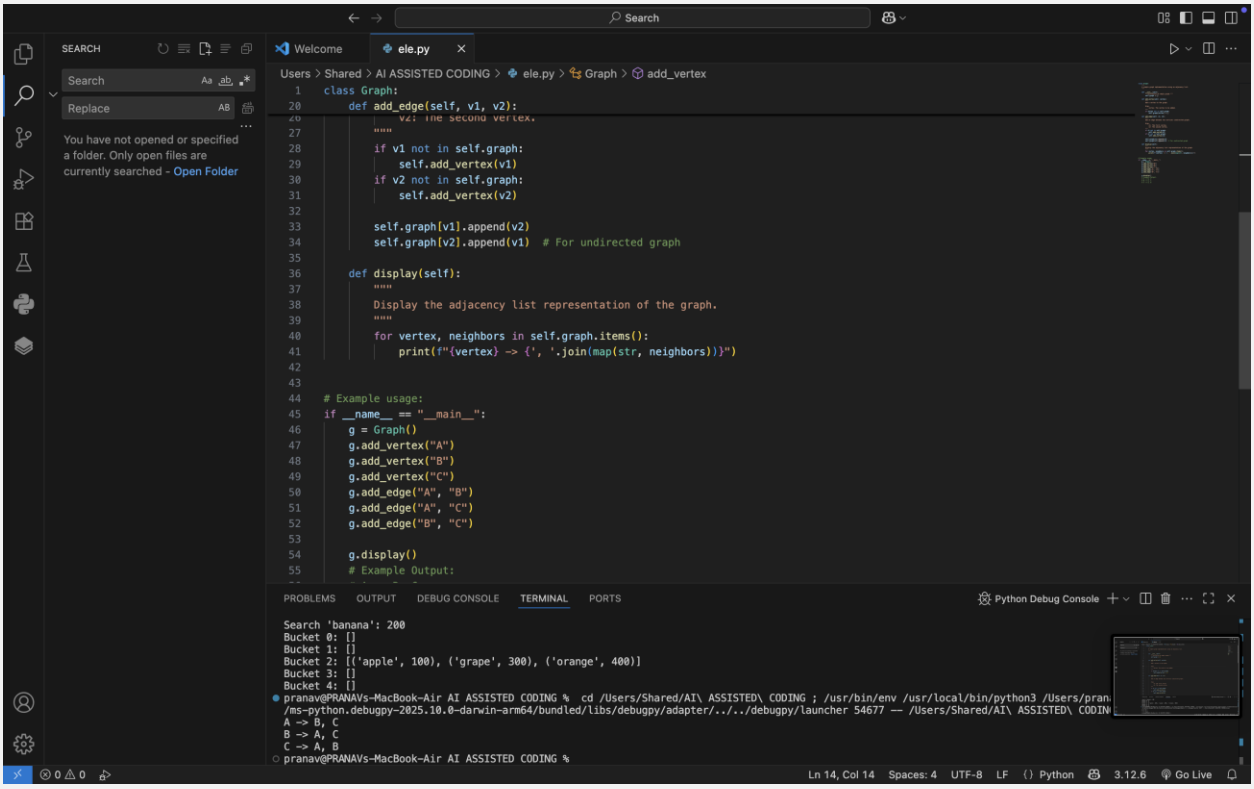

PROMPT :

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.


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
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/extensions/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 %
Ln 14, Col 14 Spaces: 4 UTF-8 LF Python 3.12.6 Go Live


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
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 %
Ln 14, Col 14 Spaces: 4 UTF-8 LF Python 3.12.6 Go Live

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/pranav/.vscode/extensions
/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 %
Ln 14, Col 14   Spaces: 4   UTF-8   LF   {} Python   3.12.6   Go Live
```

Task Description #7 – Priority Queue

Task: Use AI to implement a priority queue using Python's heapq module.

Sample Input Code:

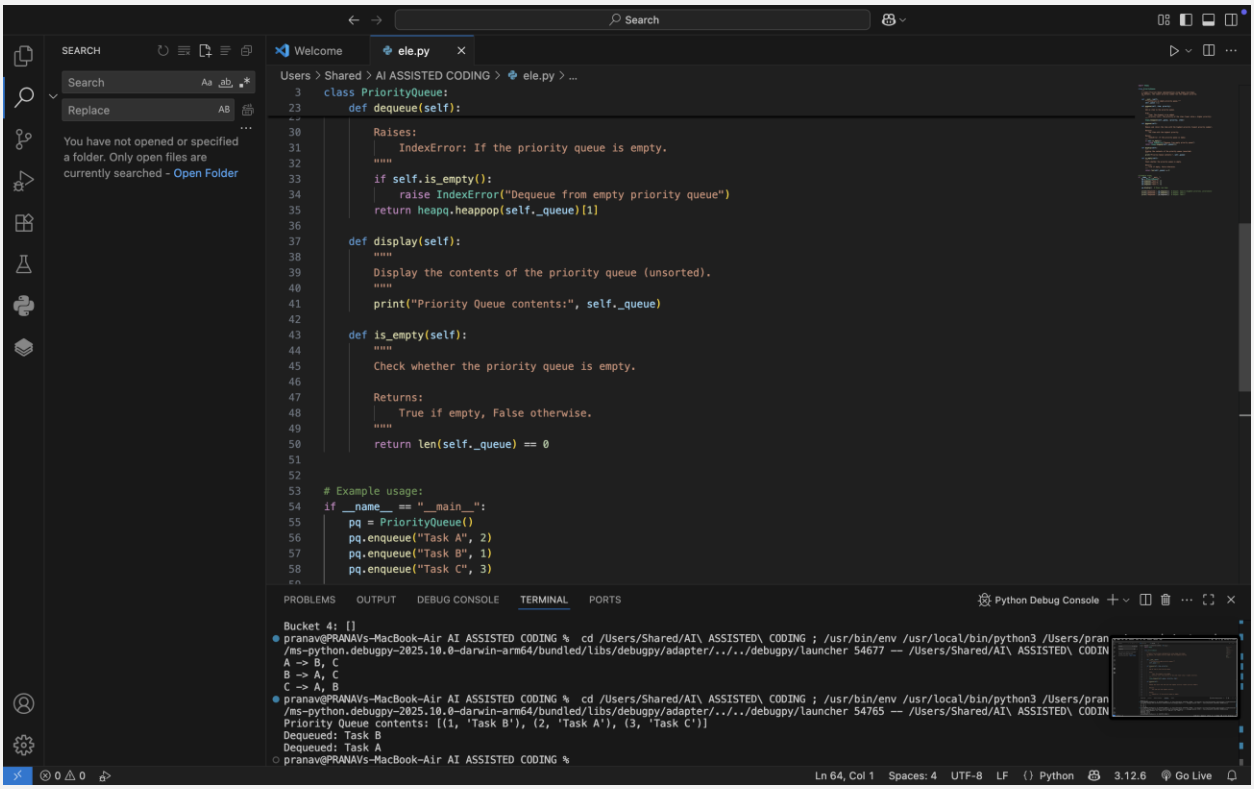
class PriorityQueue:

pass

Expected Output:

• Implementation with enqueue (priority), dequeue (highest priority), and display methods

```
Code   File   Edit   Selection   View   Go   Run   Terminal   Window   Help                Mon Sep 15  2:25 PM

Search
Replace
You have not opened or specified a folder. Only open files are currently searched - Open Folder

Users > Shared > AI ASSISTED CODING > ele.py > ...

import heapq

class PriorityQueue:
    """
    A simple Priority Queue implementation using heapq (min-heap).
    By default, the lowest priority number has the highest priority.
    """

    def __init__(self):
        """Initialize an empty priority queue."""
        self._queue = []

    def enqueue(self, item, priority):
        """
        Add an item to the priority queue.

        Args:
            item: The element to be added.
            priority (int): The priority of the item (lower value = higher priority).
        """
        heapq.heappush(self._queue, (priority, item))

    def dequeue(self):
        """
        Remove and return the item with the highest priority (lowest priority number).

        Returns:
            The item with the highest priority.

        Raises:
            IndexError: If the priority queue is empty.
        """

PROBLEMS   OUTPUT   DEBUG CONSOLE   TERMINAL   PORTS                                   Python Debug Console

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/extensions
/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 %  cd /Users/Shared/AI\ ASSISTED\ CODING ; /usr/bin/env /usr/local/bin/python3 /Users/pranav/.vscode/extensions
/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 %
Ln 64, Col 1   Spaces: 4   UTF-8   LF   {} Python   3.12.6   Go Live
```




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

Search

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 > ...

```python
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/extensions
/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/extensions
/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 %
```

Ln 86, Col 1 Spaces: 4 UTF-8 LF {} Python 3.12.6 Go Live

Search

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 > ...

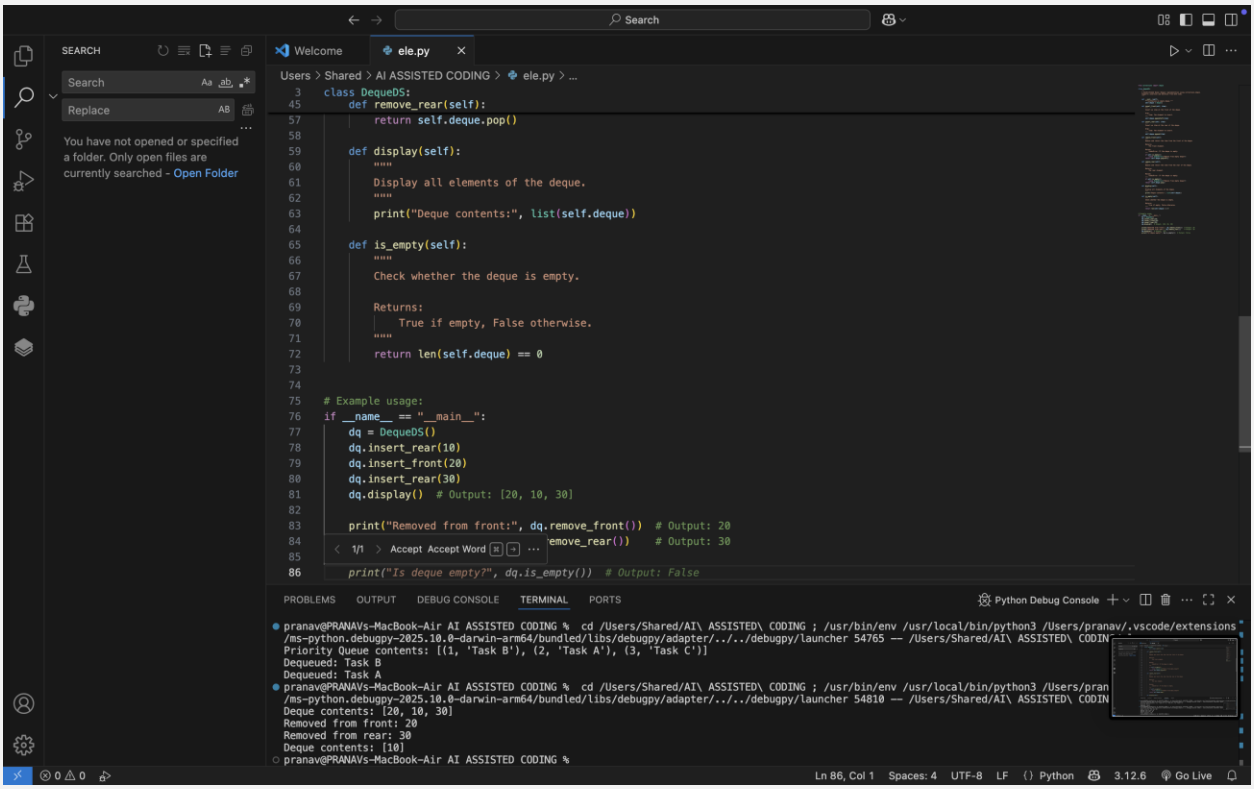
```python
class DequeDS:
        self.deque.appendleft(item)

    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/extensions
/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 %
```

Ln 86, Col 1 Spaces: 4 UTF-8 LF {} Python 3.12.6 Go Live

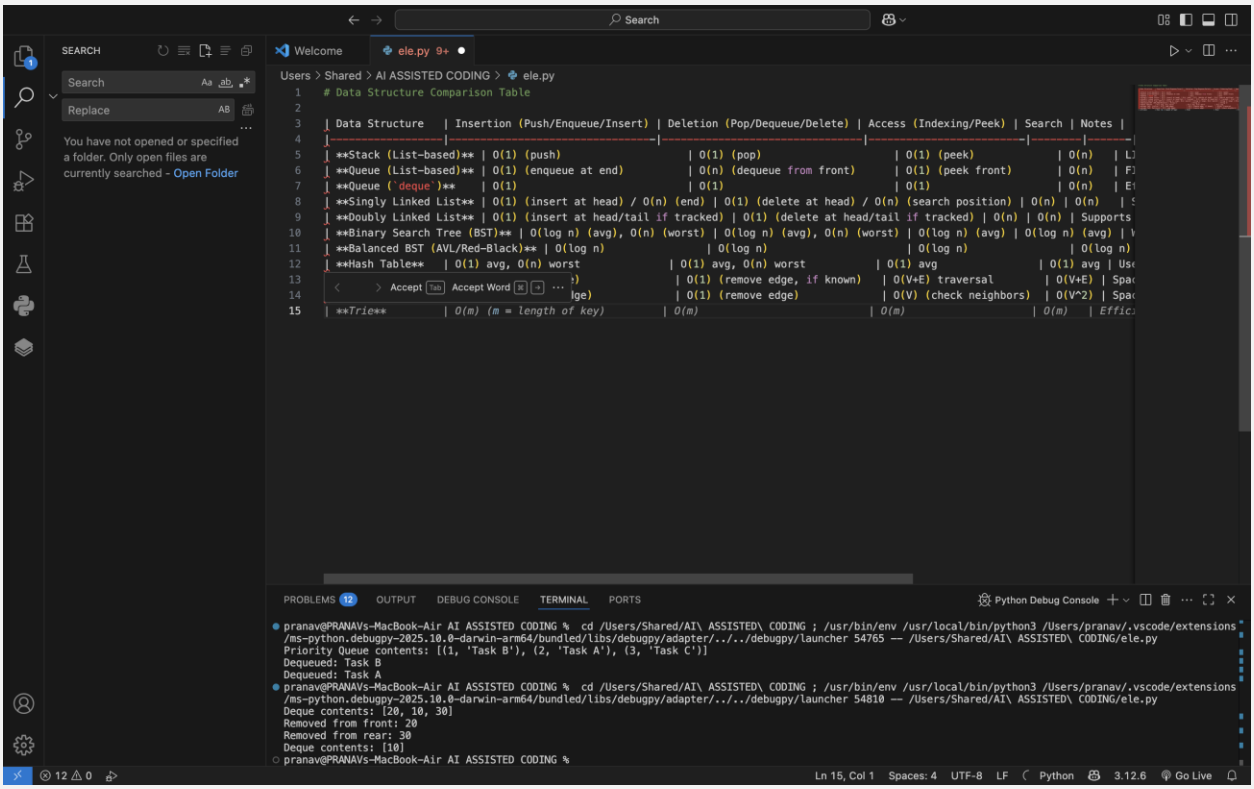


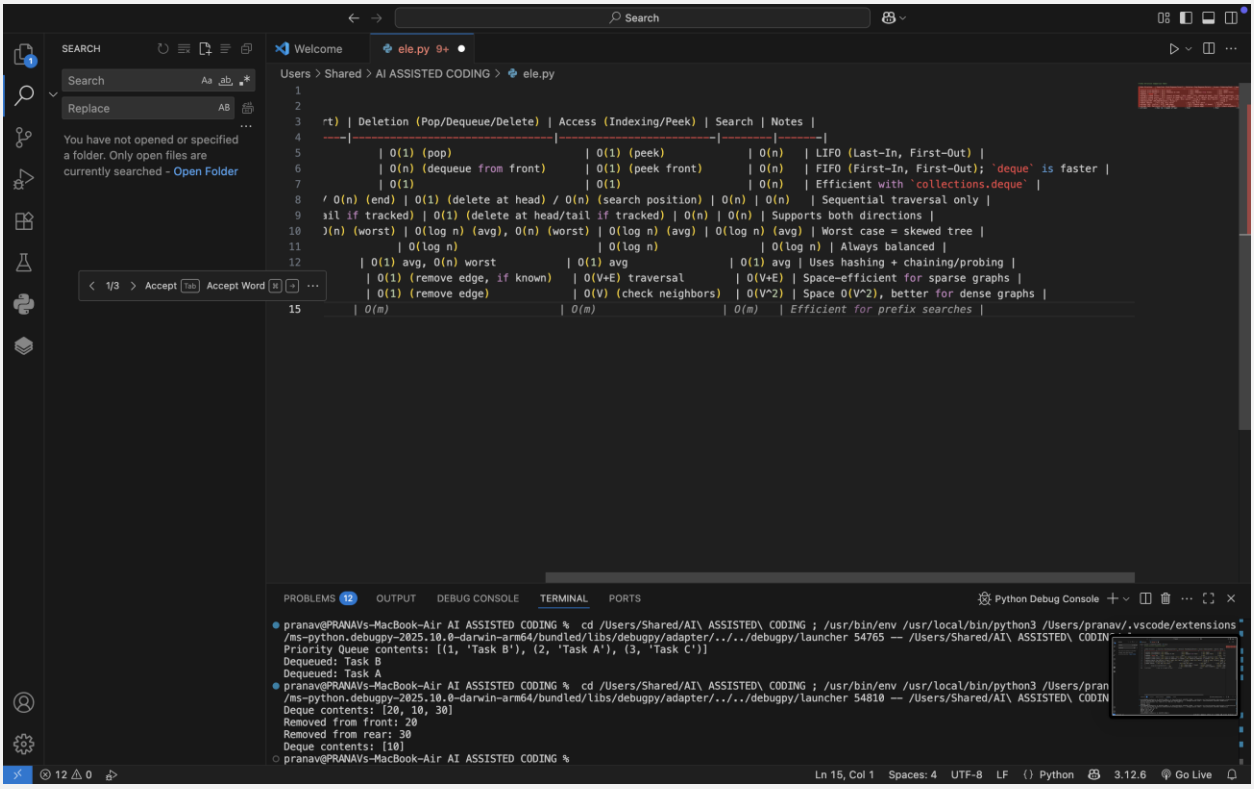

OUTPUT:


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


Search
SEARCH
Search
Replace
You have not opened or specified a folder. Only open files are currently searched - Open Folder
Welcome
ele.py 9+
Users > Shared > AI ASSISTED CODING > ele.py
# Data Structure Comparison Table
| Data Structure | Insertion (Push/Enqueue/Insert) | Deletion (Pop/Dequeue/Delete) | Access (Indexing/Peek) | Search | Notes |
| **Stack (List-based)** | O(1) (push) | O(1) (pop) | O(1) (peek) | O(n) |
| **Queue (List-based)** | O(1) (enqueue at end) | O(n) (dequeue from front) | O(1) (peek front) | O(n) |
| **Queue (`deque`)** | O(1) | O(1) | O(1) | O(n) |
| **Singly Linked List** | O(1) (insert at head) / O(n) (end) | O(1) (delete at head) / O(n) (search position) | O(n) | O(n) |
| **Doubly Linked List** | O(1) (insert at head/tail if tracked) | O(1) (delete at head/tail if tracked) | O(n) | O(n) | Supports |
| **Binary Search Tree (BST)** | O(log n) (avg), O(n) (worst) | O(log n) (avg), O(n) (worst) | O(log n) (avg) | O(log n) (avg) |
| **Balanced BST (AVL/Red-Black)** | O(log n) | O(log n) | O(log n) | O(log n) |
| **Hash Table** | O(1) avg, O(n) worst | O(1) avg, O(n) worst | O(1) avg | O(1) avg |
| | | O(1) (remove edge, if known) | O(V+E) traversal | O(V+E) |
| | | O(1) (remove edge) | O(V) (check neighbors) | O(V^2) |
| **Trie** | O(m) (m = length of key) | O(m) | O(m) | O(m) |
Accept Tab Accept Word
PROBLEMS 12 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/extensions/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/extensions/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 %
Ln 15, Col 1 Spaces: 4 UTF-8 LF Python 3.12.6 Go Live


Search
SEARCH
Search
Replace
You have not opened or specified a folder. Only open files are currently searched - Open Folder
Welcome
ele.py 9+
Users > Shared > AI ASSISTED CODING > ele.py
| Deletion (Pop/Dequeue/Delete) | Access (Indexing/Peek) | Search | Notes |
| O(1) (pop) | O(1) (peek) | O(n) | LIFO (Last-In, First-Out) |
| O(n) (dequeue from front) | O(1) (peek front) | O(n) | FIFO (First-In, First-Out); `deque` is faster |
| O(1) | O(1) | O(n) | Efficient with `collections.deque` |
| O(n) (end) | O(1) (delete at head) / O(n) (search position) | O(n) | O(n) | Sequential traversal only |
| if tracked) | O(1) (delete at head/tail if tracked) | O(n) | O(n) | Supports both directions |
| (worst) | O(log n) (avg), O(n) (worst) | O(log n) (avg) | O(log n) (avg) | Worst case = skewed tree |
| O(log n) | O(log n) | O(log n) | Always balanced |
| O(1) avg, O(n) worst | O(1) avg | O(1) avg | Uses hashing + chaining/probing |
| O(1) (remove edge, if known) | O(V+E) traversal | O(V+E) | Space-efficient for sparse graphs |
| O(1) (remove edge) | O(V) (check neighbors) | O(V^2) | Space O(V^2), better for dense graphs |
| O(m) | O(m) | O(m) | Efficient for prefix searches |
1/3 Accept Tab Accept Word
PROBLEMS 12 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/extensions/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 %
Ln 15, Col 1 Spaces: 4 UTF-8 LF Python 3.12.6 Go Live

PROMPT :

To generate a comparison table of different data structures (stack, queue, linked list, etc.) including time complexities

Task Description #10 Real-Time Application Challenge – Choose the Right Data Structure

Scenario:

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.

Expected Output:

- A table mapping feature → chosen data structure → justification.
- A functional Python program implementing the chosen feature with comments and docstrings

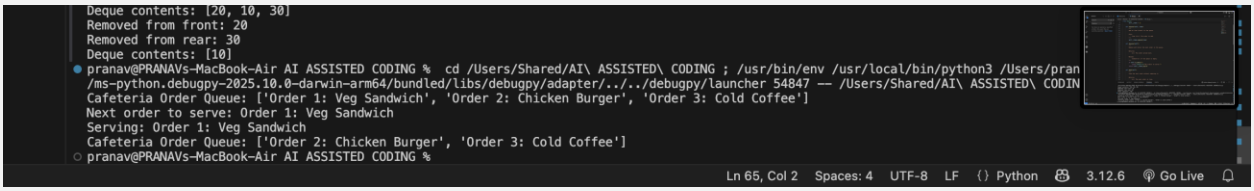
```python
class Queue:
    """
    A simple Queue implementation (FIFO - First In, First Out).
    Used here for cafeteria order management.
    """

    def __init__(self):
        """Initialize an empty queue."""
        self._items = []

    def enqueue(self, item):
        """
        Add an item (order) to the queue.

        Args:
            item (str): The order to add.
        """
        self._items.append(item)

    def dequeue(self):
        """
        Remove and return the next order in the queue.

        Returns:
            str: The order served next.

        Raises:
            IndexError: If the queue is empty.
        """
        if self.is_empty():
            raise IndexError("No orders to serve.")
        return self._items.pop(0)
```

```
/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 %  cd /Users/Shared/AI\ ASSISTED\ CODING ; /usr/bin/env /usr/local/bin/python3 /Users/pranav/.vscode/extensions
/ms-python.debugpy-2025.10.0-darwin-arm64/bundled/libs/debugpy/adapter/../../debugpy/launcher 54847 -- /Users/Shared/AI\ ASSISTED\ CODING/ele.py
Cafeteria Order Queue: ['Order 1: Veg Sandwich', 'Order 2: Chicken Burger', 'Order 3: Cold Coffee']
Next order to serve: Order 1: Veg Sandwich
Serving: Order 1: Veg Sandwich
Cafeteria Order Queue: ['Order 2: Chicken Burger', 'Order 3: Cold Coffee']
pranav@PRANAVs-MacBook-Air AI ASSISTED CODING %
```

```python
class Queue:
        self._items = []

    def enqueue(self, item):
        """
        Add an item (order) to the queue.

        Args:
            item (str): The order to add.
        """
        self._items.append(item)

    def dequeue(self):
        """
        Remove and return the next order in the queue.

        Returns:
            str: The order served next.

        Raises:
            IndexError: If the queue is empty.
        """
        if self.is_empty():
            raise IndexError("No orders to serve.")
        return self._items.pop(0)

    def peek(self):
        """
        View the next order without removing it.

        Returns:
            str: The next order in line.
```

```
/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 %  cd /Users/Shared/AI\ ASSISTED\ CODING ; /usr/bin/env /usr/local/bin/python3 /Users/pranav/.vscode/extensions
/ms-python.debugpy-2025.10.0-darwin-arm64/bundled/libs/debugpy/adapter/../../debugpy/launcher 54847 -- /Users/Shared/AI\ ASSISTED\ CODING/ele.py
Cafeteria Order Queue: ['Order 1: Veg Sandwich', 'Order 2: Chicken Burger', 'Order 3: Cold Coffee']
Next order to serve: Order 1: Veg Sandwich
Serving: Order 1: Veg Sandwich
Cafeteria Order Queue: ['Order 2: Chicken Burger', 'Order 3: Cold Coffee']
pranav@PRANAVs-MacBook-Air AI ASSISTED CODING %
```

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.