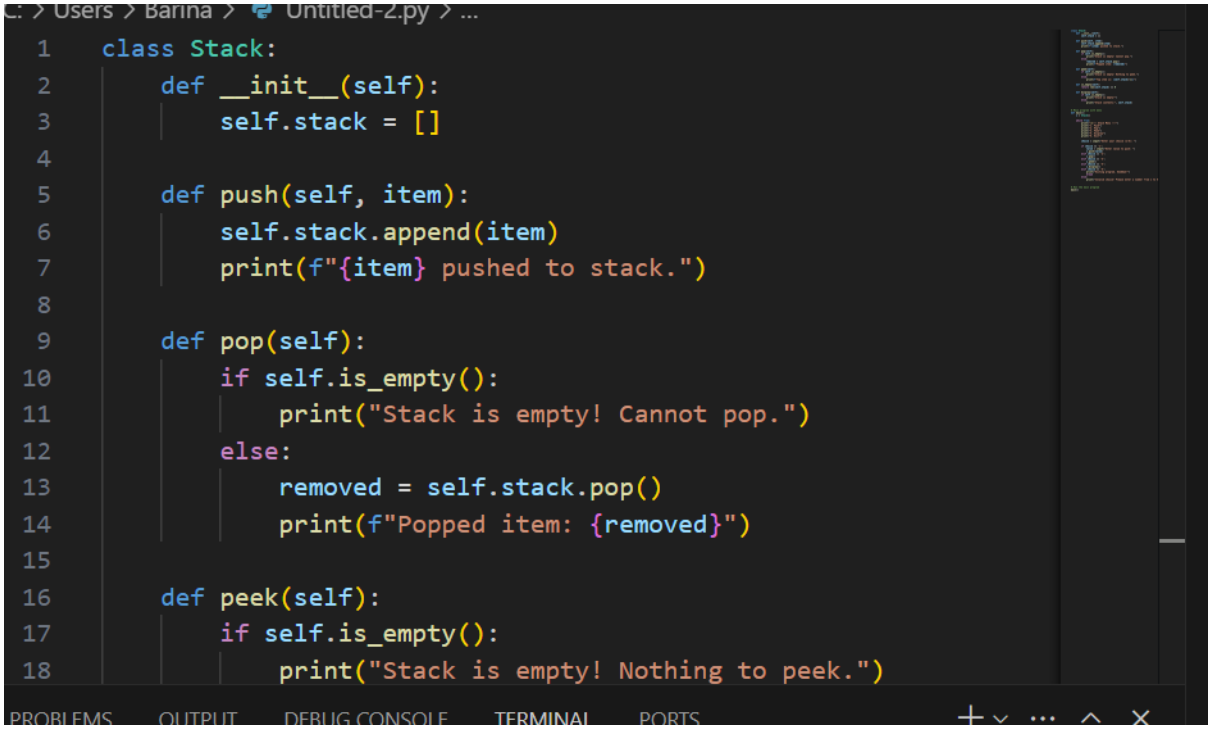
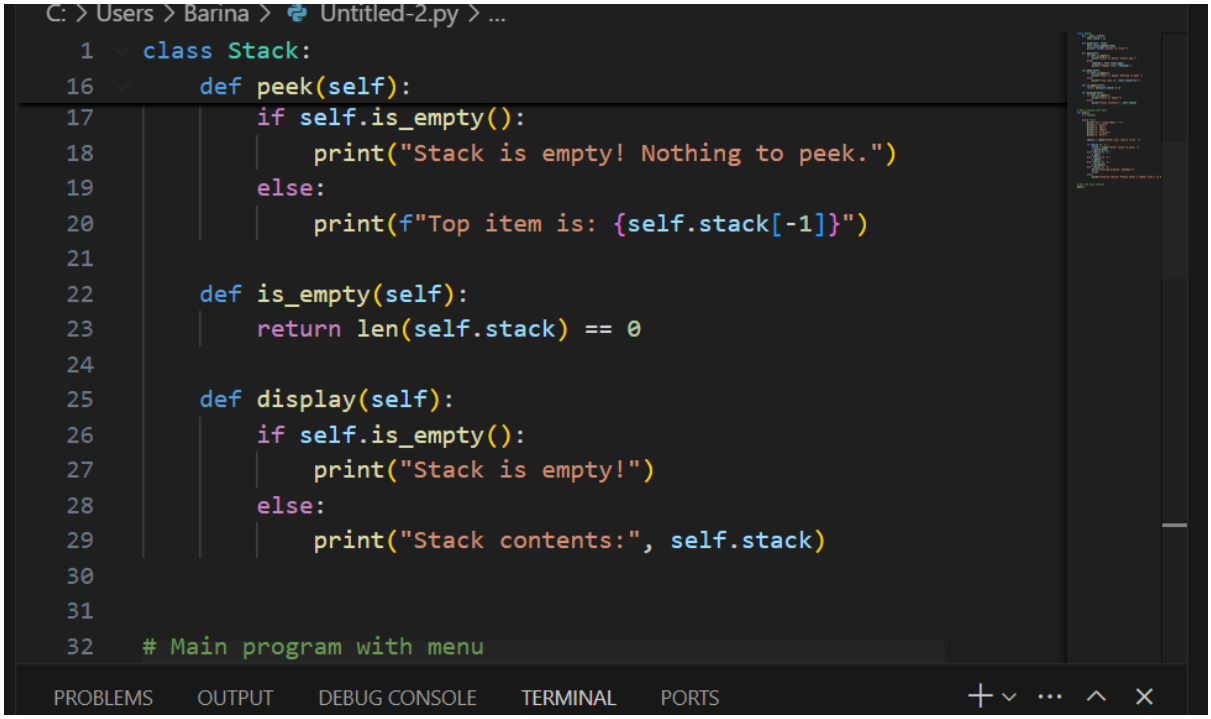
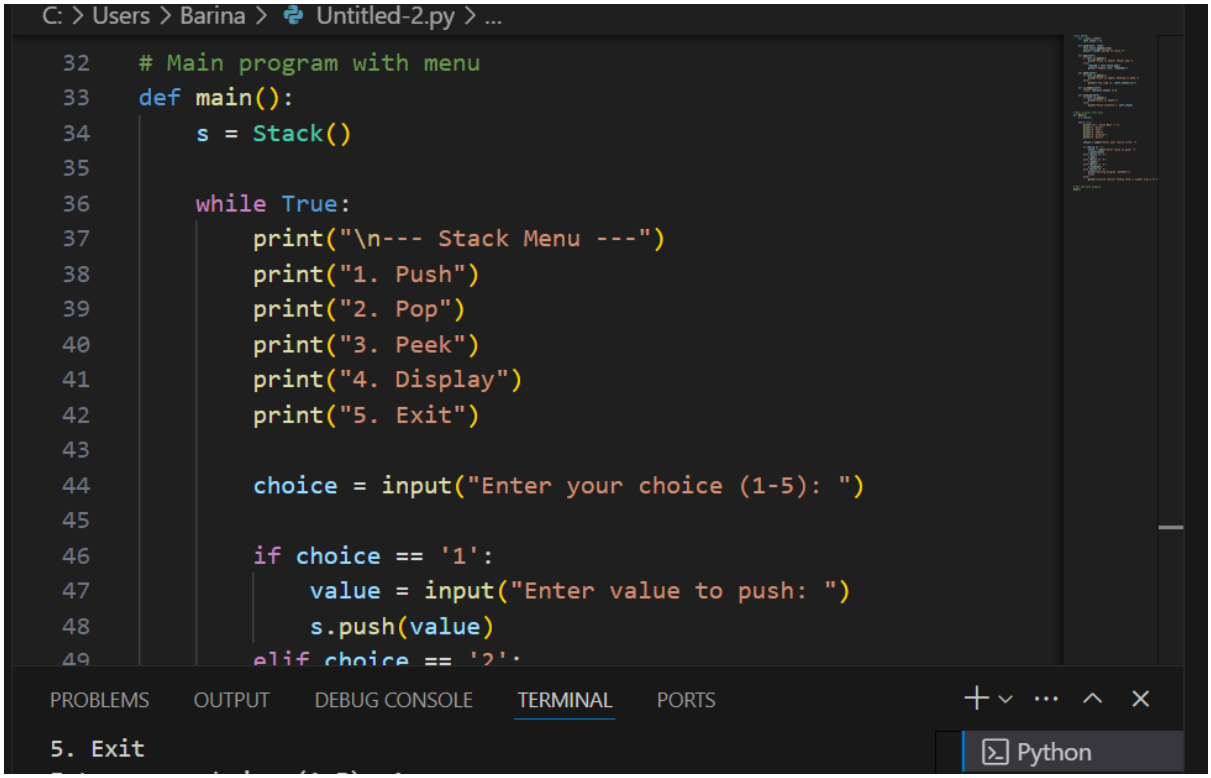
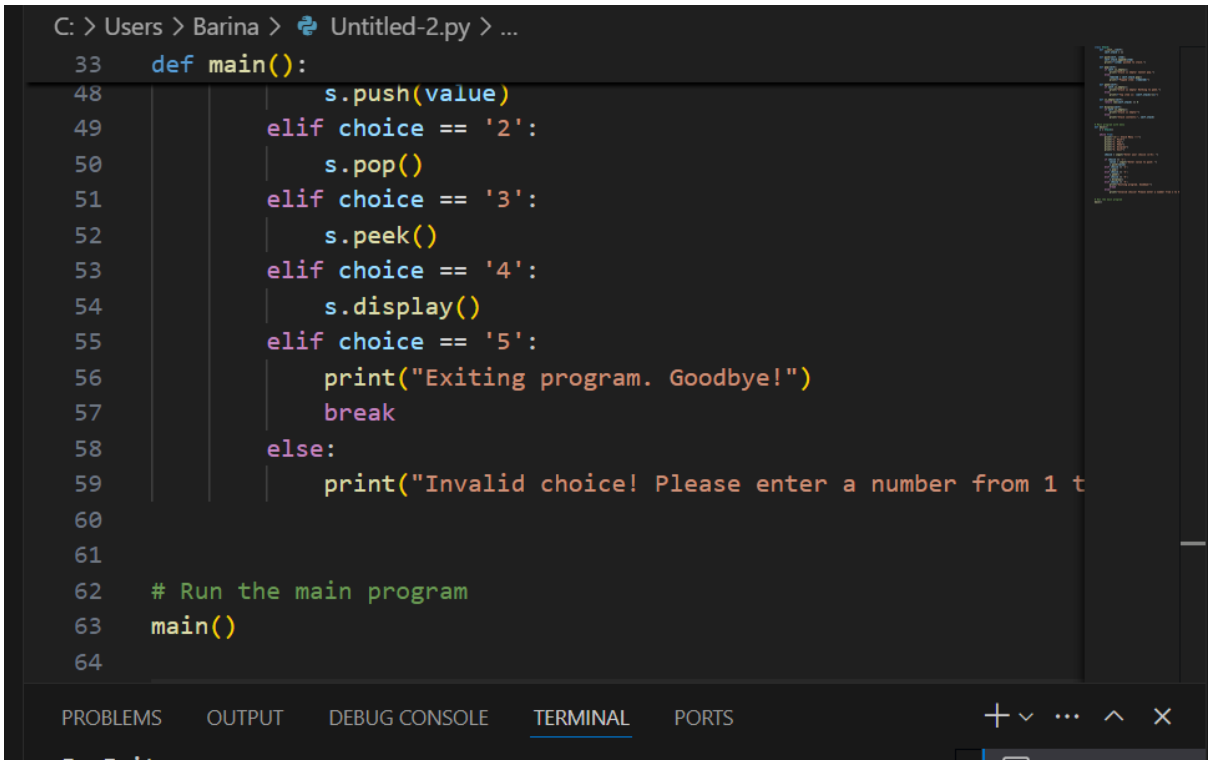
Question:

1 Implement Stack using Python.









Question:

2 Implement Queue using Python.

C: > Users > Barina > Untitled-2.py > ...

```python
class Queue:
    def __init__(self):
        self.queue = []  # empty list to store queue elements

    def enqueue(self, item):
        self.queue.append(item)
        print(f"{item} added to the queue.")

    def dequeue(self):
        if self.is_empty():
            print("Queue is empty! Cannot dequeue.")
        else:
            removed = self.queue.pop(0)  # remove the first item
            print(f"Dequeued item: {removed}")

    def peek(self):
        if self.is_empty():
            print("Queue is empty! Nothing to peek.")
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

5. Exit

Python

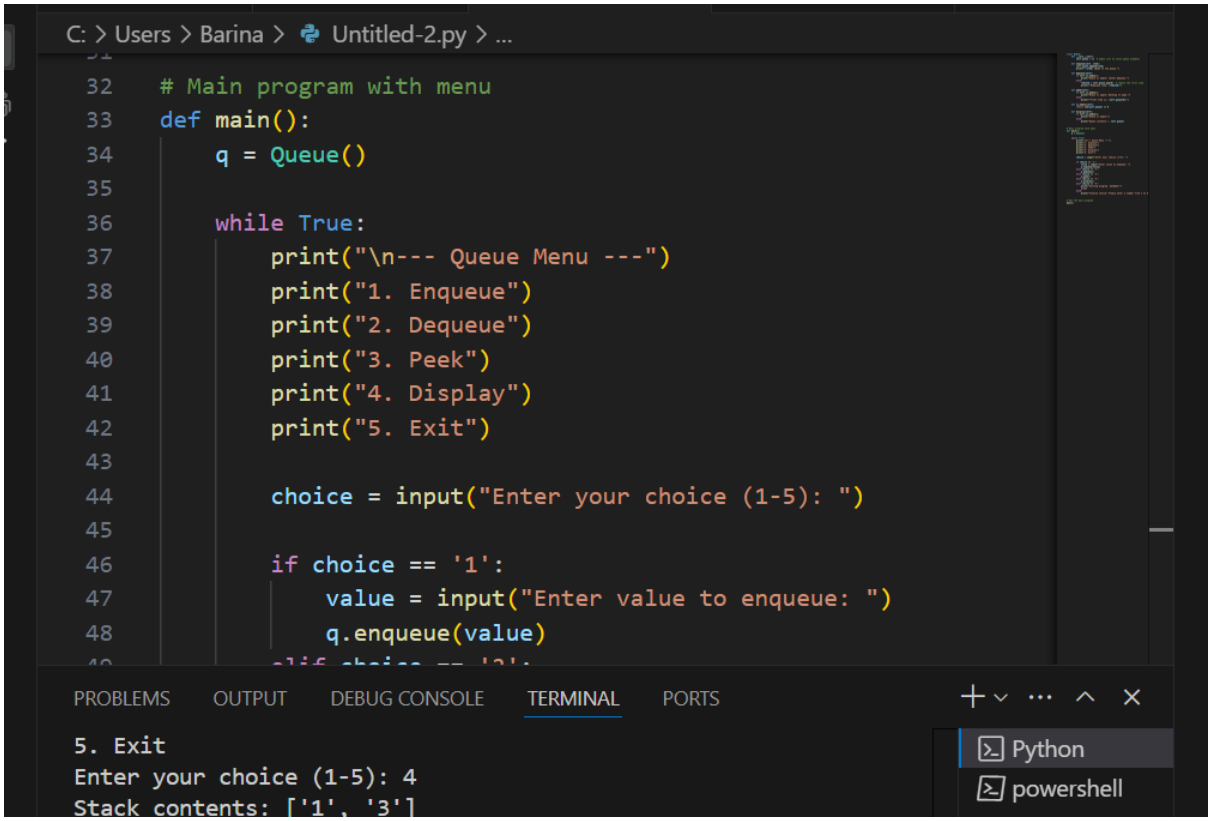
C: > Users > Barina > Untitled-2.py > ...

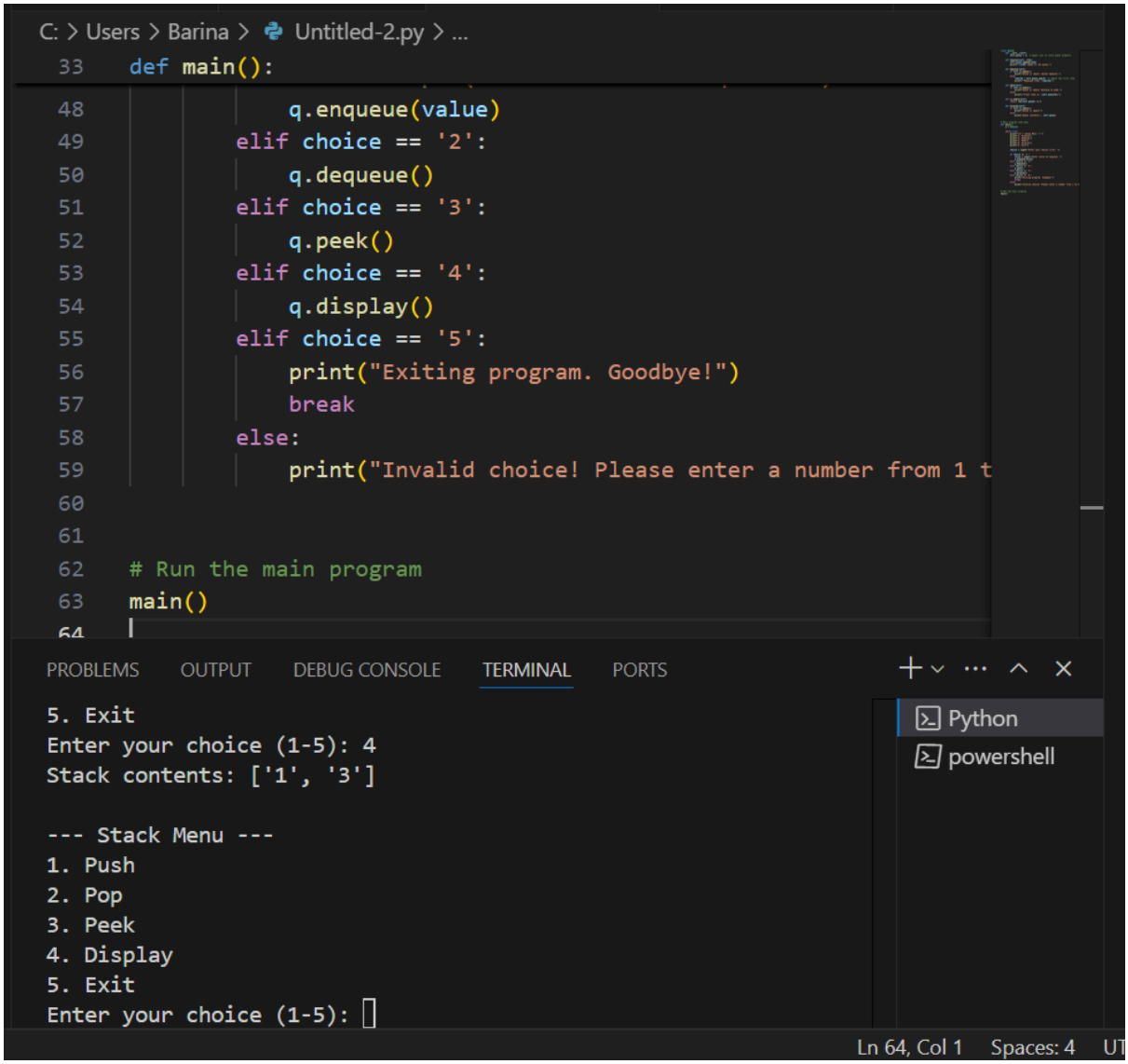
```python
class Queue:
    def peek(self):
        if self.is_empty():
            print("Queue is empty! Nothing to peek.")
        else:
            print(f"Front item is: {self.queue[0]}")

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

    def display(self):
        if self.is_empty():
            print("Queue is empty!")
        else:
            print("Queue contents:", self.queue)


# Main program with menu
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

5. Exit

Python


C: > Users > Barina > Untitled-2.py > ...
32 # Main program with menu
33 def main():
34 q = Queue()
35
36 while True:
37 print("\n--- Queue Menu ---")
38 print("1. Enqueue")
39 print("2. Dequeue")
40 print("3. Peek")
41 print("4. Display")
42 print("5. Exit")
43
44 choice = input("Enter your choice (1-5): ")
45
46 if choice == '1':
47 value = input("Enter value to enqueue: ")
48 q.enqueue(value)
PROBLEMS
OUTPUT
DEBUG CONSOLE
TERMINAL
PORTS
Python
powershell
5. Exit
Enter your choice (1-5): 4
Stack contents: ['1', '3']



Question:3 In computer science, a binary search or half-interval search algorithm finds the position of a target value within a sorted array. The binary search algorithm can be classified as a dichotomies divide-and-conquer search algorithm and executes in logarithmic time.

C: > Users > Barina > Untitled-2.py > ...

```python
def binary_search(nums, element):
    nums.sort()  # sort the list
    start = 0
    end = len(nums) - 1

    while start <= end:
        mid = (start + end) // 2

        if nums[mid] == element:
            return True
        elif nums[mid] > element:
            end = mid - 1
        else:
            start = mid + 1

    return False
nu = [1, 2, 3, 4, 5, 6, 7, 8]

```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

Welcome python.py Untitled-2.py Untitled-1.py

C: > Users > Barina > Untitled-2.py > ...

```python
def binary_search(nums, element):
            end = mid - 1
        else:
            start = mid + 1

    return False
nu = [1, 2, 3, 4, 5, 6, 7, 8]

print(binary_search(nu, 10))   # → False
print(binary_search(nu, 5))    # → True
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

```
PS C:\Users\Barina> & "C:/Program Files/Python313/python.exe" c:/Users/Barina/Untitled-2.py
False
True
PS C:\Users\Barina>
```

Python
Python