SIZE = 5

queue = [None] \* SIZE

front = -1

rear = -1

def enqueue(value):

global rear, front

if rear == SIZE - 1:

print("Queue is FULL! Insertion is not possible!")

else:

if front == -1:

front = 0

rear += 1

queue[rear] = value

print(f"{value} enqueued to queue.")

def dequeue():

global rear, front

if front == -1 or front > rear:

print("Queue is EMPTY! Cannot

dequeue.")

else:

removed = queue[front]

print(f"{removed} dequeued from queue.")

front += 1

if front > rear:

front = rear = -1

def display():

if front == -1 or front > rear:

print("Queue is EMPTY!")

else:

print("Queue elements are:")

for i in range(front, rear + 1):

print(queue[i])

while True:

print("\n--- Queue Operations Menu ---")

print("1. Enqueue")

print("2. Dequeue")

print("3. Display")

print("4. Exit")

choice = input("Enter your choice (1-4): ")

if choice == '1':

value = input("Enter value to enqueue: ")

enqueue(value)

elif choice == '2':

dequeue()

elif choice == '3':

display()

elif choice == '4':

print("Exiting program. Goodbye!")

break

else:

print("Invalid choice. Please try again.")

--- Queue Operations Menu ---

1. Enqueue

2. Dequeue

3. Display

4. Exit

Enter your choice (1-4): 1

Enter value to enqueue: 10

10 enqueued to queue.

--- Queue Operations Menu ---

1. Enqueue

2. Dequeue

3. Display

4. Exit

Enter your choice (1-4): 1

Enter value to enqueue: 20

20 enqueued to queue.

--- Queue Operations Menu ---

1. Enqueue

2. Dequeue

3. Display

4. Exit

Enter your choice (1-4): 1

Enter value to enqueue: 30

30 enqueued to queue.

--- Queue Operations Menu ---

1. Enqueue

2. Dequeue

3. Display

4. Exit

Enter your choice (1-4): 3

Queue elements are:

10

20

30

--- Queue Operations Menu ---

1. Enqueue

2. Dequeue

3. Display

4. Exit

Enter your choice (1-4): 2

10 dequeued from queue.

--- Queue Operations Menu ---

1. Enqueue

2. Dequeue

3. Display

4. Exit

Enter your choice (1-4): 4

Exiting program. Goodbye!