

### Circular Queue:

```
size = int(input("Enter size of queue: "))
queue = [None] * size
head = tail = -1

def is_empty():
    return head == -1

while True:
    print("1. Enqueue")
    print("2. Dequeue")
    print("3. Exit (-1)")
    choice = int(input("Enter your choice: "))

    if choice == -1:
        break

    if choice == 1:
        ele = int(input("Enter element to insert: "))
        if (tail + 1) % size == head:
            print("Queue is full")
        elif head == -1:
            head = tail = 0
            queue[tail] = ele
        else:
            tail = (tail + 1) % size
            queue[tail] = ele
    elif choice == 2:
        if is_empty():
            print("Queue is empty. Cannot dequeue.")
        else:
            temp = queue[head]
            if head == tail:
                head = tail = -1
            else:
                head = (head + 1) % size
            print("Dequeued element:", temp)
    else:
        print("Invalid choice")

# Print the elements in the queue
if not is_empty():
    for i in range(head, (tail + 1) % size):
        print(queue[i], end=' ')
    print()
else:
    print("Queue is empty.")
```

# Linked list implementation in Python

```
class Node:
```

```
    def __init__(self, item):
        self.item = item
        self.next = None
```

```
class LinkedList:
```

```
    def __init__(self):
        self.head = None
```

```
if __name__ == '__main__':
```

```
    linked_list = LinkedList()
```

```
    linked_list.head = Node(1)
    second = Node(2)
    third = Node(3)
```

```
    linked_list.head.next = second
    second.next = third
```

```
    while linked_list.head != None:
        print(linked_list.head.item, end=" ")
        linked_list.head = linked_list.head.next
```