**UCS 2312 Data Structures Lab**

**Assignment 5: QueueADT and its applications**

Create an ADT for the circular queue data structure with the following functions. queue*ADT* will

have the integer array, front and rear. [CO1, K3]

1. createQueue(Q) – initialize size and front and rear with -1
2. enqueue(Q,data) – push data into the queue if queue is not full.
3. Dequeue(Q) – returns the data if queue is not empty. otherwise, the function returns -1
4. isEmpty(Q) – returns 1 if queue empty, otherwise returns 0
5. isFull(Q) – returns 1 if queue full, otherwise returns 0

Test the operations of queueADT with the following test cases

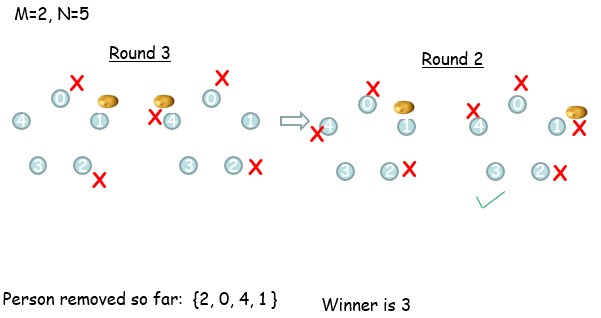
|  |  |
| --- | --- |
| **Operation** | **Expected Output** |
| createQueue(Q, 2) |  |
| enqueue(Queue,10) | 10 |
| enqueue(Queue,20) | 20 |
| enqueue(Queue,30) | Full |
| dequeue(Queue) | 10 |
| dequeue(Queue) | 20 |
| dequeue(Queue) | Empty |

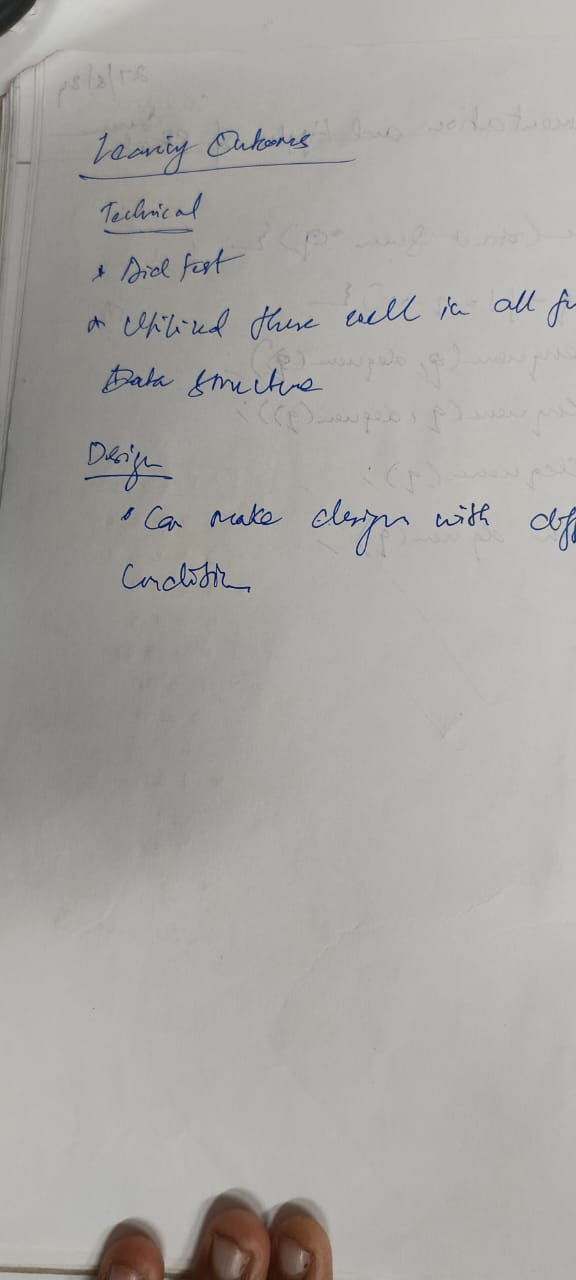
**Best practices to be followed:**

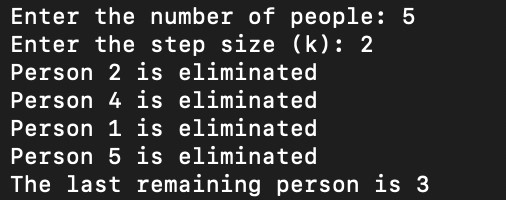
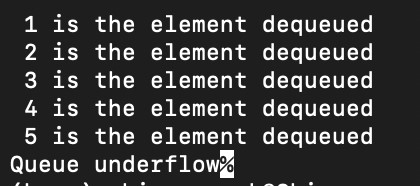
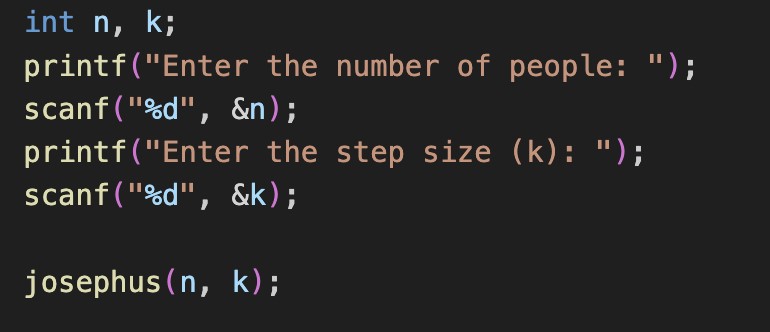
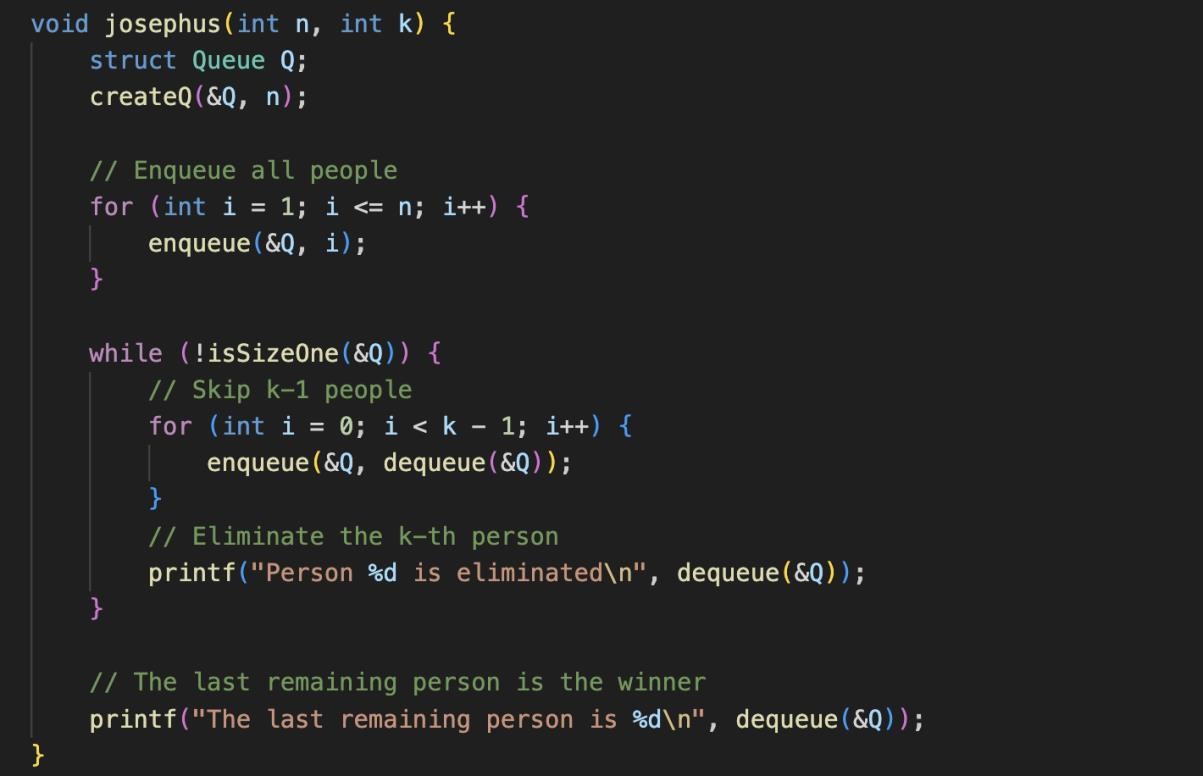
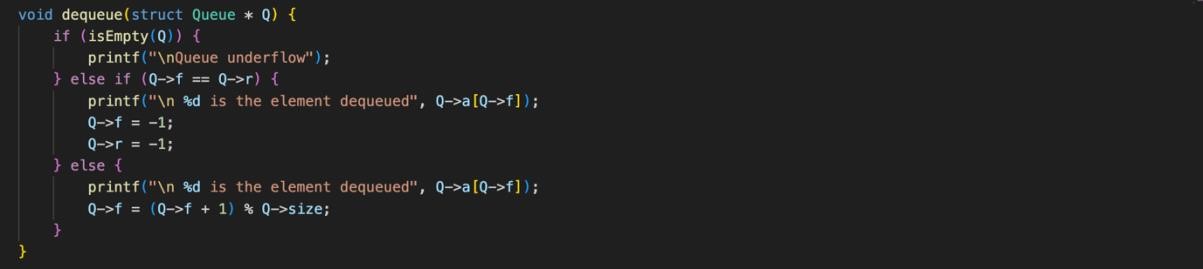
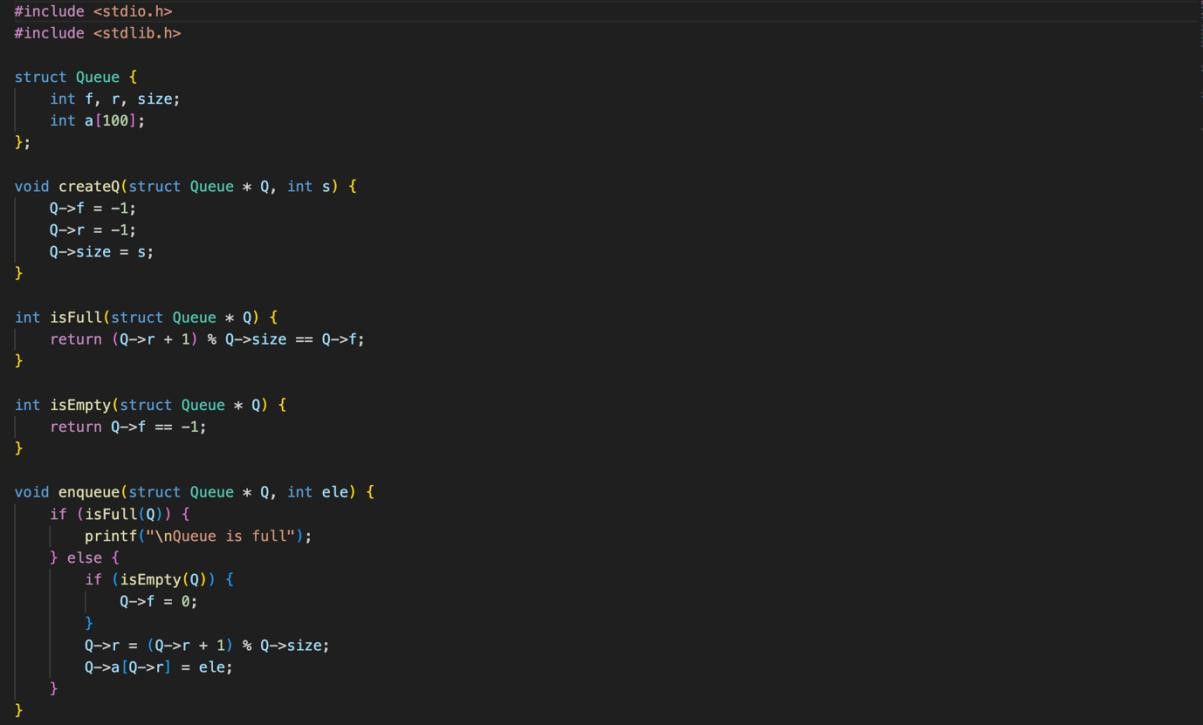
* Design before coding
* Usage of algorithm notation
* Use of multi-file C program
* Versioning of code

**Application using Queue**

Implement an **integer Circular Queue ADT** and driver to use the queue. Write an algorithm to play **Hot Potato Game** to find the winner.







Technical Outcomes

|  |  |  |
| --- | --- | --- |
| Design | 2 | Needs improvement |
| Understanding of DS | 3 |  |
| Use of DS | 3 |  |
| Debugging | 3 |  |

Best Practices

|  |  |  |
| --- | --- | --- |
| Design before coding | 2 | Needs improvement |
| Usage of Algo | 3 |  |
| Multifile | 3 |  |
| Versioning | 3 |  |