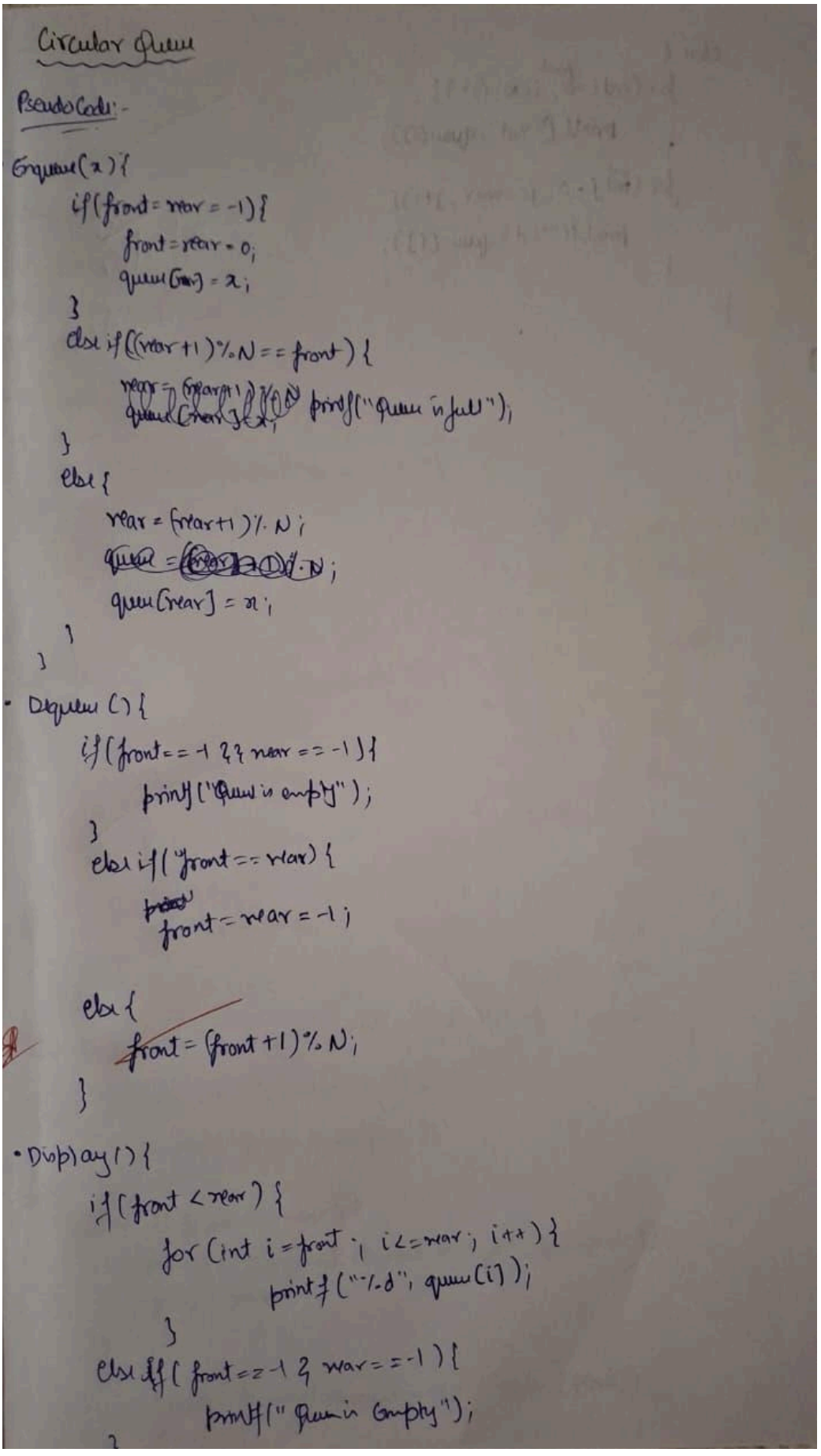
OUTPUT:

```
Enter choice:
1.Insert
2.Delete
3.Display
4.Exit
1
Enter element to insert:10

Enter choice:
1.Insert
2.Delete
3.Display
4.Exit
1
Enter element to insert:20

Enter choice:
1.Insert
2.Delete
3.Display
4.Exit
1
Enter element to insert:30
```

```
Enter choice:
1.Insert
2.Delete
3.Display
4.Exit
1
Enter element to insert:40

Enter choice:
1.Insert
2.Delete
3.Display
4.Exit
1
Enter element to insert:50

Enter choice:
1.Insert
2.Delete
3.Display
4.Exit
3
10  20  30  40  50
Enter choice:
1.Insert
2.Delete
3.Display
4.Exit
2
```

```
Dequeued element is 10
Enter choice:
1.Insert
2.Delete
3.Display
4.Exit
3
20  30  40  50
Enter choice:
1.Insert
2.Delete
3.Display
4.Exit
1
Enter element to insert:60

Enter choice:
1.Insert
2.Delete
3.Display
4.Exit
3
60  20  30  40  50
Enter choice:
1.Insert
2.Delete
3.Display
4.Exit
4
```

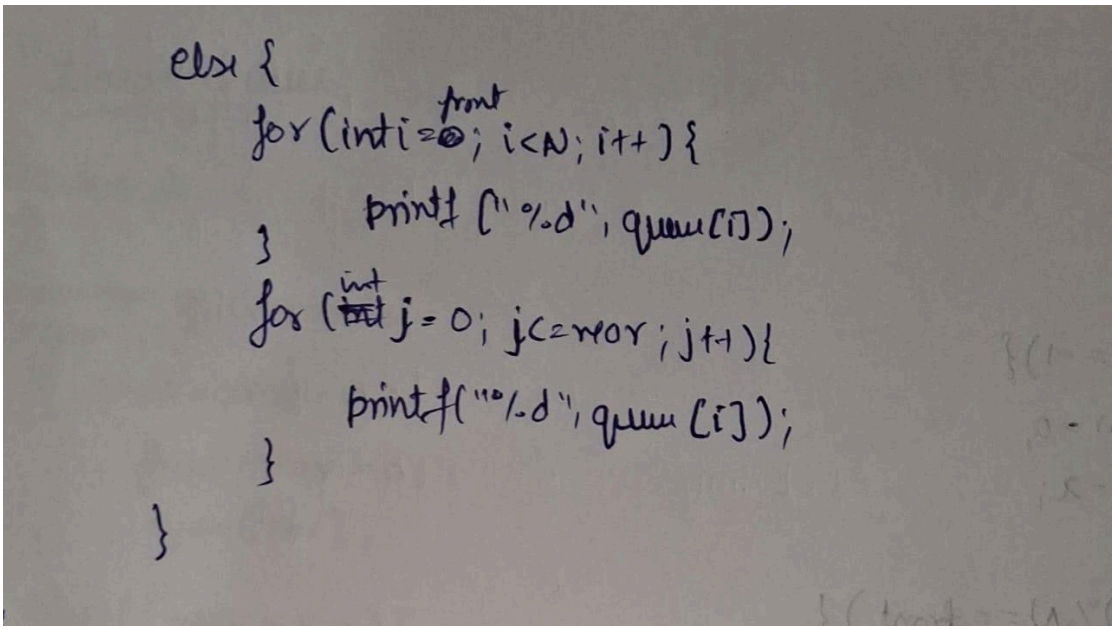
OBSERVATION:



Circular Queue

Pseudo Code:-

```
Enqueue(x){
    if(front = rear = -1){
        front = rear = 0;
        queue[rear] = x;
    }
    else if((rear+1)%N == front){
        printf("Queue is full");
    }
    else{
        rear = (rear+1)%N;
        queue[rear] = x;
    }
}

• Dequeue(){
    if(front == -1 && rear == -1){
        printf("Queue is empty");
    }
    else if(front == rear){
        front = rear = -1;
    }
    else{
        front = (front+1)%N;
    }
}

• Display(){
    if(front < rear){
        for(int i = front; i <= rear; i++){
            printf("%d", queue[i]);
        }
    else if(front == -1 && rear == -1){
        printf("Queue is Empty");
    }
```


else {
for(int i=0; i<N; i++){
printf("%d", queue[i]);
}
for(int j=0; j<=rear; j++){
printf("%d", queue[i]);
}
}
front

Circular Queue

Code →

```
#include <stdio.h>

#define N 5

int queue[N];
int front=-1;
int rear=-1;

int enqueue(){
    int x;
    printf("Enter element to insert:");
    scanf("%d",&x);
    if(front==-1&&rear==-1){
        front=rear=0;
        queue[rear]=x;
    }
    else if((rear+1)%N==front){
        printf("queue is full");
    }
    else{
        rear=(rear+1)%N;
        queue[rear]=x;
    }
}
int dequeue(){
    if(front==-1&&rear==-1){
        printf("queue is empty");
    }
    else if(front==rear){
        printf("Dequeued element is %d",queue[front]);
        front=rear=-1;
    }
    else{
        printf("Dequeued element is %d",queue[front]);
        front=(front+1)%N;
}}
```

```
void display() {
    if (front == rear == -1) {
        printf("Queue is empty");
    }
    else if (front < rear) {
        for (int i=0; i<= rear; i++) {
            printf("%d ", queue[i]);
        }
    }
    else {
        for (int i=0; i<N; i++) {
            printf("%d", queue[i]);
        }
        for (int j=0; j<=rear; j++) {
            printf("%d ", queue[j]);
        }
    }
}
Int main() {
    int choice;
    printf("Enter choice: \n1. Insert \n2. Delete \n3. Display \n4. Exit");
    Scanf("%d", &choice);
    Switch (choice) {
        case 1: Enqueue();
        break;
        case 2: Dequeue();
        break;
        case 3: Display();
        break;
        Case 4: break;
        default: printf("Invalid choice");
    }
}
```