Question:
Design and implement a given type of (ordinary queue, circular queue) queue in C (array implementation/ Linked list implementation). And demonstrate its working with suitable
inputs. Display appropriate messages in case of exceptions.

Aim:
To Implement Linear Queue using Arrays

Algorithm:

Enqueue:

- Firstly take the input from the user for the value to be inserted
- Check if the queue is full using the rear and front pointers
- If the queue is not full, check if it's empty
- If it is empty set both front and rear to 0
- Else increment the rear pointer
- Finally change the value at rear pointer to the item which the user intended to insert

Dequeue:

- Firstly check if the queue is empty using the front and rear pointers
- If it's empty, throw an error saying that the queue is empty and trying to remove an element is the Underflow condition
- Then check if the queue only has 1 element (this can be done by checking if front and rear pointers are equal)
- If it does indeed have only 1 element, then set the front and rear pointers to -1 indicating that the queue is now empty
- Finally if it doesn't satisfy the above conditions then increment front by 1

Display

- Firstly check if the queue is empty using the front and rear pointers
- If it is empty, then display a message saying that the queue is empty
- Else display all the elements starting from front to rear using a for loop

Program

```c
#include<stdio.h>
#include<stdlib.h>
#define max 100
int front=-1,rear=-1;
int queue[max];
void enqueue()
{
    int item;
    printf("Enter the element\n"); //Inserting an element
    scanf("%d",&item);
    if(rear==max-1)
    {
        printf("overflow\n");
        return;
    }
    if(front==-1&&rear==-1)
    {
        front=0;
        rear=0;
    }
    else
    {
        rear=rear+1;
    }
    queue[rear]=item;
    printf("Value inserted\n");
}
void dequeue() // to delete an element from the queue
{
    int item;
    if(front==-1||front>rear)
    {
        printf("underflow\n");
        return;
    }
    else
    {
        item=queue[front];
```

```
            if(front==rear)
            {
                front=-1;
                rear=-1;
            }
            else
            {
                front=front+1;
            }
            printf("Value deleted\n");
        }
    }
    void display()//displays all the elements
    {
        int i;
        if(rear==-1)
        {
            printf("Empty queue\n");
        }
        else
        {
            printf("The values are:\n");
            for(i=front;i<=rear;i++)
            {
                printf("%d\n",queue[i]);
            }
        }
    }
    int main()//main function
    {
        int choice;
        while(1)
        {
            printf("Select the operation:\n");
                printf("1.Enqueue\n2.Dequeue\n3.Display\n4.Exit\n");
                scanf("%d",&choice);
                switch(choice)
                {
                case 1: enqueue();
                break;
```

```
                    case 2: dequeue();
                        break;
                    case 3: display();
                        break;
                    case 4: exit(0);
                    default: printf("Invalid choice\n");
                        break;
                }
            }
            return 0;
        }
```

Output

```
Select the operation:
1.Enqueue
2.Dequeue
3.Display
4.Exit
1
Enter the element
2
Value inserted
Select the operation:
1.Enqueue
2.Dequeue
3.Display
4.Exit
1
Enter the element
3
Value inserted
Select the operation:
1.Enqueue
2.Dequeue
3.Display
4.Exit
3
The values are:
2
3
Select the operation:
1.Enqueue
2.Dequeue
3.Display
4.Exit
The values are:
2
3
Select the operation:
1.Enqueue
2.Dequeue
3.Display
4.Exit
2
Value deleted
Select the operation:
1.Enqueue
2.Dequeue
3.Display
4.Exit
3
The values are:
3
Select the operation:
1.Enqueue
2.Dequeue
3.Display
```