

```

PS E:\Programming assignments> cd "e:\Programming assignments\DS\Queues\" ; if ($?) { gcc queue.c -o
1. Queue      2. DeQueue      3. Display queue      4. Exit
Your choice: 1
Enter the element to queue: 58
Successfully Queued
1. Queue      2. DeQueue      3. Display queue      4. Exit
Your choice: 1
Enter the element to queue: 99
Successfully Queued
1. Queue      2. DeQueue      3. Display queue      4. Exit
Your choice: 3
Queue:  58      99
End
1. Queue      2. DeQueue      3. Display queue      4. Exit
Your choice: 1
Enter the element to queue: 78
Successfully Queued
1. Queue      2. DeQueue      3. Display queue      4. Exit
Your choice: 1
Enter the element to queue: 33
Successfully Queued
1. Queue      2. DeQueue      3. Display queue      4. Exit
Your choice: 3
Queue:  58      99      78      33
End
1. Queue      2. DeQueue      3. Display queue      4. Exit
Your choice: 1
Enter the element to queue: 5
Queue is already full
1. Queue      2. DeQueue      3. Display queue      4. Exit
Your choice: 3
Queue:  58      99      78      33
End
1. Queue      2. DeQueue      3. Display queue      4. Exit
Your choice: 2
Successfully deQueued: 58
1. Queue      2. DeQueue      3. Display queue      4. Exit
Your choice: 3
Queue:  99      78      33
End
1. Queue      2. DeQueue      3. Display queue      4. Exit
Your choice: 2
Successfully deQueued: 99
1. Queue      2. DeQueue      3. Display queue      4. Exit
Your choice: 1
Enter the element to queue: 7
Queue is already full
1. Queue      2. DeQueue      3. Display queue      4. Exit
Your choice: 3
Queue:  78      33
End
1. Queue      2. DeQueue      3. Display queue      4. Exit
Your choice: 4
PS E:\Programming assignments\DS\Queues>

```

```

int main()
{
    int choice, el;
    while (1){
        printf("1. Queue\t");
        printf("2. DeQueue\t");
        printf("3. Display queue \t");
        printf("4. Exit \n");
        printf("Your choice: ");
        scanf("%d",&choice);
        switch (choice)
        {
            case 1:
                printf("Enter the element to queue: ");
                scanf("%d",&el);
                queueEl(el);
                break;
            case 2:
                dequeue();
                break;
            case 3:
                display();
                break;
            case 4:
                return 1;
                break;
            default:
                printf("Invalid choice opted! \n");
                break;
        }

    }

    return 0;
}

```

Queue

```

#include<stdio.h>

#define size 4
int queue[size];
int front = -1, rear = -1;

void queueEl(int element)
{
    if(rear==size-1 || front>rear)
        printf("Queue is already full\n");
    else if(front==-1 && rear==-1){
        front++;
        rear++;
        queue[rear] = element;
        printf("Successfully Queued\n");
    }
    else{
        rear++;
        queue[rear]=element;
        printf("Successfully Queued\n");
    }
}

void dequeue()
{
    if(( front==-1 && rear==-1 ) || front>rear)
        printf("There are no elements in the queue to dequeue\n");
    else{
        printf("Successfully deQueued: %d\n", queue[front]);
        front++;
    }
}

void display()
{
    if(( front==-1 && rear==-1 ) || front>rear)
        printf("Empty queue\n");
    else{
        int traverse = front;
        printf("Queue:\t");
        while (traverse<=rear){
            printf("%d\t",queue[traverse]);
            traverse++;
        }
        printf("\nEnd\n");
    }
}

```

Queue

```

#include<stdio.h>
#define size 4
int queue[size];
int front = -1, rear = -1;
void queueEl(int element){
    if( (rear==size-1 && front==0) || (rear==front-1) )
        printf("Queue is already full\n");
    else if(front==-1 && rear==-1){
        front++;
        rear++;
        queue[rear] = element;
        printf("Successfully Queued\n");
    }
    else if(rear == size-1){
        rear = 0;
        queue[rear] = element;
    }
    else{
        rear++;
        queue[rear] = element;
    }
}

void dequeue(){
    int del;
    if(( front==-1 && rear==-1 ) )
        printf("There are no elements in the queue to dequeue\n");
    else{
        if(front==rear){
            del = queue[front];
            front = -1;
            rear = -1;
        }
        else if(front==size-1){
            del = queue[front];
            front=0;
        }
        else{
            del=queue[front];
            front++;
        }
        printf("Successfully deQueued: %d\n", del);
    }
}

void display(){
    int traverse;
    if(( front==-1 && rear==-1 ))
        printf("Empty queue\n");
}

```

Circular Queue
----------------

```

else if(front<=rear){
    traverse = front;
    printf("Queue:\t");
    for(int i=front; i<=rear; i++)
        printf("%d\t",queue[i]);
}
else{
    printf("Queue:\t");
    for(int i=front; i<size; i++)
        printf("%d\t",queue[i]);
    for(int i=0; i<=rear; i++)
        printf("%d\t",queue[i]);

}
printf("\nEnd\n");

}
int main(){
    int choice, el;
    while (1){
        printf("1. Queue\t");
        printf("2. DeQueue\t");
        printf("3. Display queue\t");
        printf("4. Exit\n");
        printf("Your choice: ");
        scanf("%d",&choice);
        switch (choice){
            case 1:
                printf("Enter the element to queue: ");
                scanf("%d",&el);
                queueEl(el);
                break;
            case 2:
                dequeue();
                break;
            case 3:
                display();
                break;
            case 4:
                return 1;
                break;

            default:
                printf("Invalid choice opted! \n");
                break;
        }
    }
    return 0;
}

```

Circular Queue
----------------

```

PS E:\Programming assignments> cd "e:\Programming assignments\DS\Queues\" ; if ($?) { gcc circularQueue.c -o ci
$?) { .\circularQueue }
1. Queue      2. DeQueue      3. Display queue      4. Exit
Your choice: 1
Enter the element to queue: 54
Successfully Queued
1. Queue      2. DeQueue      3. Display queue      4. Exit
Your choice: 1
Enter the element to queue: 77
1. Queue      2. DeQueue      3. Display queue      4. Exit
Your choice: 1
Enter the element to queue: 9
1. Queue      2. DeQueue      3. Display queue      4. Exit
Your choice: 3
Queue:  54      77      9
End
1. Queue      2. DeQueue      3. Display queue      4. Exit
Your choice: 1
Enter the element to queue: 8
1. Queue      2. DeQueue      3. Display queue      4. Exit
Your choice: 3
Queue:  54      77      9      8
End
1. Queue      2. DeQueue      3. Display queue      4. Exit
Your choice: 1
Enter the element to queue: 22
Queue is already full
1. Queue      2. DeQueue      3. Display queue      4. Exit
Your choice: 2
Successfully deQueued: 54
1. Queue      2. DeQueue      3. Display queue      4. Exit
Your choice: 1
Enter the element to queue: 7
1. Queue      2. DeQueue      3. Display queue      4. Exit
Your choice: 3
Queue:  77      9      8      7
End
1. Queue      2. DeQueue      3. Display queue      4. Exit
Your choice: 1
Enter the element to queue: 5
Queue is already full
1. Queue      2. DeQueue      3. Display queue      4. Exit
Your choice: 2
Successfully deQueued: 77
1. Queue      2. DeQueue      3. Display queue      4. Exit
Your choice: 999
Invalid choice opted!
1. Queue      2. DeQueue      3. Display queue      4. Exit
Your choice: 2
Successfully deQueued: 9
1. Queue      2. DeQueue      3. Display queue      4. Exit
Your choice: 2
Successfully deQueued: 8
1. Queue      2. DeQueue      3. Display queue      4. Exit
Your choice: 3
Queue:  7
End
1. Queue      2. DeQueue      3. Display queue      4. Exit
Your choice: 2
Successfully deQueued: 7
1. Queue      2. DeQueue      3. Display queue      4. Exit
Your choice: 2
There are no elements in the queue to dequeue
1. Queue      2. DeQueue      3. Display queue      4. Exit
Your choice: 4
PS E:\Programming assignments\DS\Queues> _

```

Circular Queue