```c
/*
Title- Queue Implimentation using DMA
Author- Bhakare Mahesh Santosh
ID- 492
Batch- TechnOrbit(PPA-8)
*/

#include<stdio.h>
#include<stdlib.h>

//  ---------------------- STRUCTURE DECLARATION -----------------------

struct QUEUE
{
    int data;
    struct QUEUE* next;
};

// -------------------- FUNCTION TO CREATE NODE --------------------

struct QUEUE* CreateNode()
{
    struct QUEUE* newnode = NULL;
    newnode = (struct QUEUE*)malloc(sizeof(struct QUEUE));
    if(newnode == NULL)
    {
        printf("Queue is Full.........\n");
    }
    else
    {
        printf("Enter data in Queue: ");
        scanf("%d",&(newnode->data));
        newnode->next = NULL;
    }
    return newnode;
}

// --------- FUNCTION TO INSERT ELEMENT --------------------

void enqueue(struct QUEUE** insert,struct QUEUE** delete)
{
    struct QUEUE* newnode = NULL;
    newnode = CreateNode();
    if(*insert == NULL)
    {
        *insert = *delete = newnode;
    }
    else
    {
        (*delete) -> next = newnode;
        *delete = newnode;
    }
}

// ------------------------- FUNCTION TO DELETE ELEMENT --------------------------

void dequeue(struct QUEUE** insert,struct QUEUE** delete)
{
    struct QUEUE* tempnode = *insert;
    if((*insert)->next == NULL)
    {
        printf("Deleted element is: %d\n",(*insert)->data);
        free(*insert);
        *insert = *delete = NULL;
    }
    else
```

```
    {
        *insert = (*insert) -> next;
        printf("Deleted element is: %d\n",tempnode->data);
        free(tempnode);
        tempnode = NULL;
    }
}

// ------------------------ FUNCTION TO DISPLAY QUEUE --------------------

void display(struct QUEUE* start)
{
    while(start != NULL)
    {
        printf("%d <= ",start->data);
        start = start->next;
    }
}

// ------------------- FUNCTION TO CHECK QUEUE IS EMPTY OR NOT --------------------------

int empty(struct QUEUE* insert)
{
    if(insert == NULL)
    {
        return 1;
    }
    return 0;
}

// ------------------------------ ENTRY POINT FUNCTION --------------------------

void main()
{
    struct QUEUE *insert = NULL, *delete = NULL;
    int choice;
    do
    {
        printf("*********************************************\n");
        printf("1.Enqueue\n2.Dequeue\n3.IsEmpty\n4.Display\n5.Exit\nEnter Your Choice: ");
        scanf("%d",&choice);
        switch(choice)
        {
            case 1: enqueue(&insert,&delete);
                    break;
            case 2: if(empty(insert))
                    {
                        printf("Queue Empty.....\n");
                    }
                    else
                    {
                        dequeue(&insert,&delete);
                    }
                    break;
            case 3: if(empty(insert))
                    {
                        printf("Queue is Empty....\n");
                    }
                    else
                    {
                        printf("Queue is not Empty.....\n");
                    }
                    break;
            case 4: if(empty(insert))
                    {
                        printf("Queue is Empty.....\n");
```

```
                    }
                    else
                    {
                        display(insert);
                    }
                    break;
            case 5: printf("Exiting......\n");
                    break;
            default: printf("Wrong Choice...! Please enter correct choice...\n");
        }
    }while(choice != 5);

}
```