QUEUE LINKED LIST

```
#include <stdio.h>
#include <stdlib.h>
struct Node{
    int data;
    struct Node *next;
};
struct Node *front=NULL;
struct Node *rear=NULL;
void enqueue(int value){
    struct Node *newNode = (struct Node *)malloc(sizeof(struct Node));
    if(newNode == NULL) {
        printf("Queue Overflow\n");
        return;
    }
    newNode->data=value;
    newNode->next=NULL;
    if(rear==NULL) {
        front=rear=newNode;
    }
          else{
        rear->next=newNode;
        rear=newNode;
    }
    printf("Enqueued: %d\n", value);
}
void dequeue(){
    if (front==NULL) {
        printf("Queue Underflow (Empty Queue)\n");
        return;
    }
    struct Node *temp = front;
    printf("Dequeued: %d\n",front->data);
    front=front->next;
    if(front==NULL)
        rear=NULL;
    free(temp);
}
void display(){
    if(front==NULL) {
        printf("Queue is empty\n");
        return;
    }
    struct Node *temp = front;
    printf("Queue elements: ");
    while(temp!=NULL){
        printf("%d ", temp->data);
        temp = temp->next;
    }
```

```
    printf("NULL\n");
}
int main(){
    int choice,value;
    while(1) {
        printf("\n--- Queue using Linked List ---\n");
        printf("1.Enqueue\n");
        printf("2.Dequeue\n");
        printf("3.Display\n");
        printf("4.Exit\n");
        printf("Enter your choice: ");
        scanf("%d", &choice);
        switch(choice){
        case 1:
            printf("Enter value to enqueue: ");
            scanf("%d", &value);
            enqueue(value);
            break;
        case 2:
            dequeue();
            break;
        case 3:
            display();
            break;
        case 4:
            printf("Exiting...\n");
            exit(0);
        default:
            printf("Invalid choice! Try again.\n");
        }
    }
    return 0;
}
```

OUTPUT:

```
1.Enqueue
2.Dequeue
3.Display
4.Exit
Enter your choice: 1
Enter value to enqueue: 3
Enqueued: 3

--- Queue using Linked List ---
1.Enqueue
2.Dequeue
3.Display
4.Exit
Enter your choice: 2
Dequeued: 1

--- Queue using Linked List ---
1.Enqueue
2.Dequeue
3.Display
4.Exit
Enter your choice: 3
Queue elements: 3 NULL

--- Queue using Linked List ---
1.Enqueue
2.Dequeue
3.Display
4.Exit
Enter your choice:
```