```c
#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 * ptr;
    ptr = (struct node * ) malloc(sizeof(struct node));
    ptr -> data = value;
    ptr -> next = NULL;
    if ((front == NULL) && (rear == NULL)) {
        front = rear = ptr;
    } else {
        rear -> next = ptr;
        rear = ptr;
    }
    printf("Node is Inserted\n\n");
}

int dequeue() {
    if (front == NULL) {
        printf("\nUnderflow\n");
        return -1;
    } else {
        struct node * temp = front;
        int temp_data = front -> data;
        front = front -> next;
        free(temp);
        return temp_data;
    }
}

void display() {
    struct node * temp;
    if ((front == NULL) && (rear == NULL)) {
        printf("\nQueue is Empty\n");
    } else {
```

```
        printf("The queue is \n");
        temp = front;
        while (temp) {
            printf("%d--->", temp -> data);
            temp = temp -> next;
        }
        printf("NULL\n\n");
    }
}

int main() {
    int choice, value;
    printf("\nImplementation of Queue using Linked List\n");
    while (choice != 4) {
        printf("1.Enqueue\n2.Dequeue\n3.Display\n4.Exit\n");
        printf("\nEnter your choice : ");
        scanf("%d", & choice);
        switch (choice) {
            case 1:
                printf("\nEnter the value to insert: ");
                scanf("%d", & value);
                enqueue(value);
                break;
            case 2:
                printf("Popped element is :%d\n", dequeue());
                break;
            case 3:
                display();
                break;
            case 4:
                exit(0);
                break;
            default:
                printf("\nWrong Choice\n");
        }
    }
    return 0;
}
```

```
1.Enqueue
2.Dequeue
3.Display
4.Exit
1

Enter the value to insert: 1
Node is Inserted

1.Enqueue
2.Dequeue
3.Display
4.Exit
1

Enter the value to insert: 2
Node is Inserted

1.Enqueue
2.Dequeue
3.Display
4.Exit
3
The queue is
1--->2--->NULL

1.Enqueue
2.Dequeue
3.Display
4.Exit
2
Popped element is :1
1.Enqueue
2.Dequeue
3.Display
4.Exit
3
The queue is
2--->NULL

1.Enqueue
2.Dequeue
3.Display
4.Exit
4

Process returned 0 (0x0)   execution time : 15.932 s
Press any key to continue.
```

```
void push(int eleme
int pop();
display()

int main()
{
    int choice, data;
    while(1){
```

Stacks, Queues, reverse, sort, concatenation

NL 29/1/Mo..

Emtd

SLL

SLL