Program

```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 item;
    printf("Enter the item to be inserted: ");
    scanf("%d", &item);
    struct node *newnode = (struct node*)malloc(sizeof(struct node));
    newnode->data = item;
    newnode->next = NULL;
    if(front==NULL && rear==NULL)
    {
      front=rear=newnode;
    }
    else
    {
      rear->next=newnode;
      rear=newnode;
    }
}
 void dequeue()
{
  struct node *temp=front;
  if(front==NULL)
    printf("Underflow");
  else
  {
    printf("Dequeued item %d",front->data);
    front=front->next;
    if(front==NULL)
    {
      rear=NULL;
      free(temp);
    }
  }
}

void display() {
    struct node *temp = front;
    if (front == NULL) {
        printf(" underflow\n");
    } else {
        while (temp != NULL) {
            printf("%d ", temp->data);
            temp = temp->next;
        }
        printf("\n");
    }
}
int main() {
    int op;
    char s;
```

QUEUE IMPLEMENTATION USING LINKED LIST

Aim:

To implement a queue using a linked list and perform operations on it.

Algorithm:

1. Start.

2. Create a structure with int data and struct node*next.

3. Declare struct node*front=NULL ,*rear=NULL

4. Create function void enqueue()

```
        Declare int item
        print "Enter the item to be inserted"
        scan to int item
        struct node*newnode=(struct node*)malloc(sizeof(struct node)
        newnode->date=value
        newnode->next=NULL
        if(front==rear==NULL)
                    front=rear=newnode
    else
                    rear->next=newnode
                    rear=newnode
        End if
```

5. Create function void dequeue()

```
        struct node*temp=front
        if front==NULL
                    print "Empty"
        else
                    print ''Deleted:%d'',front->data
                    front=front->next
                     if front==NULL
                                    rear=NULL
                                    free(temp)
        End if
```

6. Create function display()

```c
    do {
        printf("1. ENQUEUE\n2. DEQUEUE\n3. DISPLAY\n");
        printf("Enter your choice:");
        scanf("%d", &op);
        switch (op) {
            case 1:
                enqueue();
                break;
            case 2:
                dequeue();
                break;
            case 3:
                display();
                break;
            default:
                printf("Invalid option! Please try again.\n");
        }
        printf("Do you want to continue? (y/n): ");
        scanf(" %c", &s);
    } while (s == 'y' || s == 'Y');
    return 0;
}
```

Output

```
1. ENQUEUE
2. DEQUEUE
3. DISPLAY
Enter your choice:1
Enter the item to be inserted: 1
Do you want to continue? (y/n): Y
1. ENQUEUE
2. DEQUEUE
3. DISPLAY
Enter your choice:1
Enter the item to be inserted: 2
Do you want to continue? (y/n): Y
1. ENQUEUE
2. DEQUEUE
3. DISPLAY
Enter your choice:3
1 2
Do you want to continue? (y/n): Y
1. ENQUEUE
2. DEQUEUE
3. DISPLAY
Enter your choice:2
Dequeued item 1Do you want to continue? (y/n): Y
1. ENQUEUE
2. DEQUEUE
3. DISPLAY
Enter your choice:2
Dequeued item 2Do you want to continue? (y/n): Y
1. ENQUEUE
```

```
            struct node*temp=front
            if front==NULL
                       print "underflow"
            else
                Begin while loop when temp!=NULL
                          Print temp->data
                          temp=temp->next
                 End while
                           print "\n"
            End if
```

7. Create main function()

```
                 Declare int op
                 Declare char s
                 Start do
                       print 1.ENQUEUE 2.DEQUEUE 3.DISPLAY
                       print "Enter your choice"
                       scan to int op
                       switch(op)
                           case 1:
                               call insert()
                               break
                           case 2:
                               call  delete()
                               break
                           case 3 :
                               call display()
                               break
                           default:
                               print invalid option!please try again
                         Endswitch
                         print do you want to continue(y/n):
                         scan to int s
                    Begin while loop
                         s=='y'||s=='Y'
                    End while
               End of function
```

8. Stop

```
2. DEQUEUE
3. DISPLAY
Enter your choice:2
UnderflowDo you want to continue? (y/n):
```

Result:

Program has been executed successfully and obtained the output