```c
#include <stdio.h>
#include <stdlib.h>
struct node
{
    int data;
    struct node *next;
};
void qinsert();
void qdisplay();
void qdel();
void spush();
void spop();
void sdisplay();
struct  node *rear=NULL, *front =NULL, *top=NULL;

int main(int argc, char **argv)
{

    int choice;
    printf("Enter the choice\n1.Stack\n2.Queue\n");
    scanf("%d",&choice);

    if(choice==1)
    do
    { printf("\n1. Push \n2. Display  \n3. Pop\n");
    printf("\nEnter your choice : ");
    scanf("%d",&choice);
    switch(choice)
    {
        case 1: spush(); break;
        case 2: sdisplay();break;
        case 3: spop(); break;
        default: if(choice!=4)
                    printf("\nInvalid Input");
    }
```

```
clang-7 -pthread -lm -o main main.c
./main
Enter the choice
1.Stack
2.Queue
1

1. Push
2. Display
3. Pop

Enter your choice : 1
Enter the element
1

1. Push
2. Display
3. Pop

Enter your choice : 1
Enter the element
2

1. Push
2. Display
3. Pop

Enter your choice : 1
Enter the element
3

1. Push
2. Display
```

```c
    }

    }while(choice!=4);

    else if(choice==2)
    do
    { printf("\nQueue implementation using linked list\n");
      printf("\n1. Create \n2. Display  \n3. Delete  \n4. Exit
          \n");
      printf("\nEnter your choice : ");
      scanf("%d",&choice);
      switch(choice)
      { case 1: qinsert(); break;
        case 2: qdisplay();break;
        case 3: qdel(); break;
        default: if(choice!=4)
                    printf("\nInvalid Input");
      }
    }while(choice!=4);
}

void qinsert()
{
    struct  node *newnode;
    newnode=(struct node *) malloc(sizeof(struct node));
    printf("Enter the element:\n");
    scanf("%d",&newnode->data);
    newnode->next=NULL;

    if(rear==NULL)
    {
        rear=newnode;
        front=newnode;

    }
```

```
1. Push
2. Display
3. Pop

Enter your choice : 2
321
1. Push
2. Display
3. Pop

Enter your choice : 3
element removed is 3:
1. Push
2. Display
3. Pop

Enter your choice : 3
element removed is 2:
1. Push
2. Display
3. Pop

Enter your choice : 3
element removed is 1:
1. Push
2. Display
3. Pop

Enter your choice : 3
stack is empty
1. Push
2. Display
```

```c
    }
    else
    {
        rear->next=newnode;
        rear=newnode;
    }
}

void qdel()
{
    if(front==NULL)
    {
       printf("Queue is empty\n");return;
    }

    else
    {
        printf("Deleted ele is %d",front->data);
        if(front==rear)
        {
           printf("Queue is empty\n");
           front=NULL; rear=NULL;
        }
        else
        front=front->next;
    }
}

void qdisplay()
{
    struct node *temp;
    if(front ==NULL)
    {
        printf("Queue is empty");
```

```
clang-7 -pthread -lm -o main main.c
./main
Enter the choice
1.Stack
2.Queue
2

Queue implementation using linked list

1. Create
2. Display
3. Delete
4. Exit

Enter your choice : 1
Enter the element:
1

Queue implementation using linked list

1. Create
2. Display
3. Delete
4. Exit

Enter your choice : 1
Enter the element:
2

Queue implementation using linked list

1. Create
2. Display
```

```
        printf("Queue is empty");
        return;
    }
    temp=front;
    while (temp !=NULL)
    {
        printf("%d ",temp->data);
        temp=temp->next;
    }

}
void spush()
{
    int item;
    struct node *newnode;
    printf("Enter the element\n");
    scanf("%d",&item);

    newnode=(struct node*)malloc(sizeof(struct node));
    newnode->data=item;
    newnode->next=NULL;
    if(top==NULL)
        top=newnode;
    else
        newnode->next=top;
        top=newnode;
}
void spop()
{
    if(top==NULL)
        printf("stack is empty");
    else
    {
```

```
Queue implementation using linked list

1. Create
2. Display
3. Delete
4. Exit

Enter your choice : 1
Enter the element:
3

Queue implementation using linked list

1. Create
2. Display
3. Delete
4. Exit

Enter your choice : 2
1 2 3
Queue implementation using linked list

1. Create
2. Display
3. Delete
4. Exit

Enter your choice : 3
Deleted ele is 1
Queue implementation using linked list

1. Create
2. Display
```

```
}
void spop()
{
    if(top==NULL)
        printf("stack is empty");
    else
    {

      printf("element removed is %d:", top->data);

      top=top->next;

    }


}

void sdisplay()
{
struct node *temp;
temp=top;
if(top==NULL)
    printf("Stack is empty");
while(temp!=NULL)
{
    printf("%d",temp->data);
    temp=temp->next;
}

}
```

```
3. Delete
4. Exit

Enter your choice : 2
1 2 3
Queue implementation using linked list

1. Create
2. Display
3. Delete
4. Exit

Enter your choice : 3
Deleted ele is 1
Queue implementation using linked list

1. Create
2. Display
3. Delete
4. Exit

Enter your choice : 3
Deleted ele is 2
Queue implementation using linked list

1. Create
2. Display
3. Delete
4. Exit

Enter your choice : 3
Deleted ele is 3Queue is empty
```