link list application.c X stack and queue.c X

```c
#include <stdio.h>
#include <stdlib.h>
struct node
{
    int data;
    struct node *next;
};
void insert();
void del();
void display_queue();
void push();
void pop();
void display_stack();

struct node *top=NULL;
struct  node *rear=NULL, *front =NULL;

int main(int argc, char **argv)
{
    int choice;
    while(choice!=3)
    {
        printf("\n1. Stack \n2. Queue \n3. Quit\n");
        printf("Enter your choice:");
        scanf("%d",&choice);
        if(choice==1)
        {
            printf("\n------Stack---------\n");
            int choice1;
            while(choice1!=4)
            {
                printf("\n1. Push \n2. Pop  \n3. Display \n4. Quit\n");
                printf("\nEnter your choice : ");
                scanf("%d",&choice1);
                if(choice1==1)
```

link list application.c | stack and queue.c

```
            scanf("%d",&choice1);
            if(choice1==1)
            {
                push();
            }
            else if(choice1==2)
            {
                pop();
            }
            else if(choice1==3)
            {
                display_stack();
            }
            else if(choice1==4)
            {
                break;
            }
        }
    }
    else if(choice==2)
    {
        printf("\n------Queue---------\n");
        int choice2;
        while(choice2!=4)
        {
            printf("\n1. Insert \n2. Delete  \n3. Display \n4. Quit\n");
            printf("\nEnter your choice : ");
            scanf("%d",&choice2);
            if(choice2==1)
            {
                insert();
            }
            else if(choice2==2)
            {
                del();
            }
```

```
                else if(choice2==2)
                {
                    del();
                }
                else if(choice2==3)
                {
                    display_queue();
                }
                else if(choice2==4)
                {
                    break;
                }
            }
        }
        else if(choice==3)
            break;

    }
    return 0;
}

void push()
{
    int item;
    struct node *newnode;
    printf("Enter the element:");
    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;
```

link list application.c | stack and queue.c

```
        top=newnode;
}
void pop()
{
    if(top==NULL)
        printf("Stack is empty\n");
    else
    {
      printf("Element removed is %d\n", top->data);
      top=top->next;
    }
}

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

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

    if(rear==NULL)
    {
```

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

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

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

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

void display_queue()
{
    struct node *temp;
```

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

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

void display_queue()
{
    struct node *temp;
    if(front ==NULL)
    {
        printf("Queue is empty\n");
        return;
    }
    temp=front;
    while (temp !=NULL)
    {
        printf("%d ",temp->data);
        temp=temp->next;
    }
}
```