```c
//Lab Program 8
//stack and Queue implementation
#include <stdio.h>
#include <stdlib.h>
struct node{
    int data;
    struct node *next;
};

void insertend(struct node **headptr){
    struct node *newnode,*temp;
    int value;
    printf("Enter value: ");
    scanf("%d",&value);
    newnode = (struct node*)malloc(sizeof(struct node));
    newnode->data = value;
    newnode->next = NULL;
    temp = (*headptr);
    if(*headptr == NULL)
        (*headptr) = newnode;
    else{
        while(temp->next != NULL)
            temp = temp->next;
        temp->next = newnode;
    }
}
void deletefront(struct node **headptr){
    if((*headptr) == NULL)
        printf("The list is empty\n");
    else if((*headptr)->next == NULL)
        (*headptr) = NULL;
    else{
        (*headptr) = (*headptr)->next;
    }
}
void deleteend(struct node **headptr){
    struct node *temp;
    temp = (*headptr);
    if((*headptr) == NULL)
        printf("The list is empty\n");
    else if((*headptr)->next == NULL)
        (*headptr) = NULL;
    else{
        temp = *headptr;
```

```
        while((temp->next)->next != NULL)
            temp = temp->next;
        temp->next = NULL;
    }
}
void display(struct node *temp){
    if(temp == NULL){
        printf("The list is empty\n");
        return;
    }
    else{
        while(temp != NULL){
            printf("%d\t",temp->data);
            temp = temp->next;
        }
        printf("\n");
    }
}
int main(int argc,char **argv){
    struct node *head1=NULL,*head2=NULL;
    int choice;
    while(choice != 7){
        printf("Enter choice: 1)pushstack 2)popstack 3)displaystack 4)EnQ 5)DeQ 6)displayQ 7)exit : ");
        scanf("%d",&choice);
        switch(choice){
            case 1:insertend(&head1);break;
            case 2:deleteend(&head1);break;
            case 3:display(head1);break;
            case 4:insertend(&head2);break;
            case 5:deletefront(&head2);break;
            case 6:display(head2);break;
            case 7:
            default:exit(0);
        }
    }
}
```