MOHAMMED ABDUL HAMID
1BM19CS202
3D

LAB 7

```
#include<stdio.h>
#include <stdlib.h>
  struct node{
              intdata;
              struct nodeanext;
           };
       struct node *head1;
       struct node *head2;
 void push() { struct node *ptr;
               int new_data;
               ptr = (struct node *)malloc (sizeof (struct node));
    if(ptr==NULL){ printf("overflow\n");
               }
    else { printf(" Enter the Value to inserted :");
         scanf(" %d", &new_data);
           ptr->data =new_data;
           ptr->next = head;
             head=ptr;
               printf("Node inserted at top of slack \n"); }}
 void pop(){
        struct node *ptr;
          if(head==NULL){ printf("The list is empty"); }
        else { ptr=head;
             head=ptr->next;
             free(ptr);
             printf(" NODE DELED FROM TOP OF STACK \n"),
                 }}
 Void enqueue(){
          struct node *ptr, *temp
           int new_data;
          ptr=(struct node*)malloc (sizeof (struct node));
   printf(" Enter value to be inserted"
  scanf(" %d ", &new_data);
  ptr->data =new_data;
```

Scanned with CamScanner

```
if(head==NULL){
        ptr->next=NULL;
        head=ptr;
        printf(" Node inserted at rear Q");
else { temp=head;
        while (temp->next!=NULL){
                    temp=temp->next; }
        temp->next =ptr;
        ptr->next= NULL;
        printf(" NODE inserted at rear of Q"); }}
void dequeue(){
        struct node *ptr;
        if(head==NULL) { printf ("Empty"); }
    else { ptr=head;
            head=ptr->next;
            free(ptr);
            printf("NODE DELETED FROM FRONT OF QUEUE\n");
                }}
void display() {
        struct node= *ptr;
            ptr=head;
        if(ptr==NULL) { printf(" Empty \n"); }
    else { printf(" \n\n LIST -> ");
                while(ptr!=NULL){
                        printf(" %d ", ptr->data);
                        ptr= ptr->next;
            }}}
void sort(){
        struct node *ptr= head;
        struct node *temp =NULL;
        int i;
    if (head==NULL) {
            return;
            }
```

```
else { while (ptr != NULL){
                temp=ptr->next;
            while (temp != NULL){
        if (ptr->data -> temp-> data){
                    i = ptr->data;
            ptr->data = temp->data;
            temp->data=i;
            } temp= temp ->next;
        } ptr = ptr->next;
            }}}
void reverse(){
        struct node *prev= NULL;
        struct node *next= NULL;
        struct node *ptr=head;
    while (ptr != NULL){
                next= ptr->next;
                ptr->next = prev;
                prev =ptr;
                ptr = next;
            } head =prev;
        }

struct node *create_list(struct node *head){
        struct node *ptr, *temp;
        int i,n,new_data;
printf(" Enter number of nodes:");
scanf(" %d", &n);
 head=NULL;
  if (n==0){ return head; }
 for (i=1; i<=n; i++){
                ptr=(struct node *)malloc (sizeof (struct node));
printf("Enter the element to be inserted:");
 scanf(" %d", &new_data);
 if (head== NULL){
            ptr->next =NULL;
                head=ptr;
                }
```

```
else{ temp=head;
      while(temp->next!=NULL) {
                    temp= temp->next;
                 }

          temp->next=ptr;
           ptr->next=NULL;   }}}}


struct node * concatenated (struct node *head, struct node
  {
         struct node *ptr;
      if(head==NULL){
                head=head2;
                 return head; }

    if (head2==NULL){
                return head;}

          ptr=head;
           while(ptr->next!=NULL){ptr=ptr->next;}
          ptr->next= head;
                  return head;
                }
int main(){ int choice=0;
    while(1){
         printf("① Push ② POP ③ DISPLAY ④ENQUEE ④ DEQ
  ⑤ DISP ⑥ ");
  switch(){
          case:
          }
```