8a)
```
#include<stdio.h>
#include<stdlib.h>
void push();
void pop();
void display();
struct node
{int data;
 struct node *next;
};
struct node *top=NULL;
int main()
{ int choice;
  ~~printf("Sta~~
  do{
     printf("1.Push\n2.Pop\n3.Display\n4.Exit");
     scanf("%d",&choice);
     switch(choice)
       { case 1: push(); break;
         case 2: pop(); break;
         case 3: display(); brd;
         case 4: exit(0);
       }
     }while(choice!=4);
}
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;
}

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

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

8b)
```
#include<stdio.h>
#include<stdlib.h>
struct node
  { int data;
    struct node *next;
  };
void insert();
void display();
void del();
struct node *rear=NULL, *front=NULL;
int main()
 {
   printf("\nQueue implementation using
    linked list\n");
   int choice;
   do {
   printf("\n1. Create\n2. Display\n3. Delete\n
        4. Exit\n");
   scanf("%d", &choice);
   switch(choice)
     { case 1: insert(); break;
       case 2: display(); break;
       case 3: del(); break;
       case 4: exit(0);
     }
   }while(choice!=4);
 }
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)
  {
    rear = newnode;
    front = newnode;
  }
else
  { rear->next = newnode;
    rear = newnode;
  }
}
void del()
{ if(front == NULL)
    { printf("Queue is empty");
  else
    { printf("Deleted element is %d", front->data);
      if(front == rear)
        { front = NULL;
          rear = NULL;
        }
      else
        { front = front->next;
        }
    }
void display()
{ struct node *temp;
  if(front == NULL)
    { printf("Queue is empty"); return; }
  temp = front;
  while(temp != NULL)
    { printf("%d", temp->data);
      temp = temp->next; }}
```