8) Stack Linkedlist Program

```
#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;
do {
printf("\n<------MENU----->");
printf("\n<STACK LINKED LIST\n");
printf("\n1.PUSH \n2.DISPLAY \n3.POP \n4.Exit");
printf("\nPlease pick your choice :");
scanf("%d", &choice);
switch(choice) {
case 1: push(); break;
case 2: display(); break;
case 3: pop(); break;
}
}while(choice != 4);
}
void push() {
int item;
struct node *newnode;
printf("Enter the element:\n");
```

```
scanf("%d", &item);
newnode = (structnode*) malloc(sizeof(structnode));
newnode -> data = item;
newnode -> next = NULL;
if (top == NULL)
top = newnode;
else
newnode -> next = top;
top = newnode;
}
void pop() {
if (top == NULL) {
printf("\n STACK is EMPTY\n");
}else {
printf("\n Element removed is %d :", top -> data);
top = top -> next;
}
}
void display() {
{
struct node *cool;
cool = top;
if (top == NULL)
printf("\n STACK IS EMPTY !!!\n");
while (cool != NULL) {
printf("\t %d \t", cool -> data);
cool = cool -> next;
}
}
```

8) Queue Linked List Program

```
#include <stdio.h>
#include <stdlib.h>
struct node {
int data;
struct node *next;
};
void insert();
void display();
void delete();
struct node *rear = NULL, *front = NULL;
int main() {
int choice;
do {
printf("\n<------ MENU ------>\n");
printf("QUEUE LINKED LIST\n");
printf("\n1. Create \n2. DISPLAY \n3. Delete \n4. Exit \n");
printf("\n Enter your choice: ");
scanf("%d", &choice);
switch(choice) {
case 1: insert(); break;
case 2: display(); break;
case 3: delete(); break;
}
} while(choice != 4);
}
void insert() {
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;
}else {
rear->next = newnode;
rear = newnode;
}
}
void delete() {
if (front == NULL) {
print("\n QUEUE IS EMPTY!!!\n"); return;
}else {
printf("Deleted element is = %d", front->data);
if (front == rear) {
front = NULL;
rear = NULL;
}else
front = front->next;
}}
void display() {
struct node *cool;
if (front == NULL)
printf("Queue is EMPTY!!!\n"); return;
cool = front;
while (cool != NULL) {
printf("%d", cool->data);
cool = cool->next;
}
}
```