

PRACTICAL-8

Write a program to implement stack using linked list.

SOURCE CODE:

```
#include <stdio.h>

#include <stdlib.h>

#include <conio.h>

struct stack

{

    int data;

    struct stack *next;

};

struct stack *top = NULL;

struct stack *push (struct stack *, int);

struct stack *display (struct stack *);

struct stack *pop (struct stack *);

int peek (struct stack *);

int main (int argc, char *argv[])

{

    int val, option;

    while(1)

    {

        printf ("\n *****MAIN MENU*****");

        printf ("\n 1. PUSH");

        printf ("\n 2. POP");

        printf ("\n 3. PEEK");
```

```
printf ("\n 4. DISPLAY");
printf ("\n 5. EXIT");
printf ("\n Enter your option: ");
scanf ("%d", &option);
switch (option)
{
case 1:
printf ("\n Enter the number to be pushed on stack: ");
scanf ("%d", &val);
top = push (top, val);
break;
case 2:
top = pop (top);
break;
case 3:
val = peek (top);
if (val != -1)
printf ("\n The value at the top of stack is: %d", val); else
printf ("\n STACK IS EMPTY");
break;
case 4:
top = display (top);
break;
case 5:
exit(0);
}
}
```

```

return 0;
}

struct stack *push (struct stack *top, int val) {
    struct stack *ptr;
    ptr = (struct stack *) malloc (sizeof (struct stack)); ptr->data = val;
    if (top == NULL)
    {
        ptr->next = NULL;
        top = ptr;
    }
    else
    {
        ptr->next = top;
        top = ptr;
    }
    return top;
}

struct stack *display (struct stack *top)
{
    struct stack *ptr;
    ptr = top;
    if (top == NULL)
        printf ("\n STACK IS EMPTY");
    else
    {
        while (ptr != NULL)

```

```

{
    printf ("\n %d", ptr->data);
    ptr = ptr->next;
}

}

return top;
}

struct stack *pop (struct stack *top)
{
    struct stack *ptr;
    ptr = top;
    if (top == NULL)
        printf ("\n STACK UNDERFLOW");
    else
    {
        top = top->next;
        printf ("\n The value being deleted is: %d", ptr->data); free (ptr);
    }
    return top;
}

int peek (struct stack *top)
{
    if (top == NULL)
        return -1;
    else
        return top->data;
}

```

OUTPUT:

```
*****MAIN MENU*****
1. PUSH
2. POP
3. PEEK
4. DISPLAY
5. EXIT
Enter your option: 1

Enter the number to be pushed on stack: 10

*****MAIN MENU*****
1. PUSH
2. POP
3. PEEK
4. DISPLAY
5. EXIT
Enter your option: 1

Enter the number to be pushed on stack: 20

*****MAIN MENU*****
1. PUSH
2. POP
3. PEEK
4. DISPLAY
5. EXIT
Enter your option: 4
```

```

20
10
*****MAIN MENU*****
1. PUSH
2. POP
3. PEEK
4. DISPLAY
5. EXIT
Enter your option: 2

The value being deleted is: 20
*****MAIN MENU*****
1. PUSH
2. POP
3. PEEK
4. DISPLAY
5. EXIT
Enter your option: 3

The value at the top of stack is: 10
*****MAIN MENU*****
1. PUSH
2. POP
3. PEEK
4. DISPLAY
5. EXIT
Enter your option: █

```

PRACTICAL-9

Write a program to implement queue using linked list.

SOURCE CODE:

```

#include<stdio.h>

#include<stdlib.h>

struct node
{
    int data;
    struct node *next;
};

struct node *front;
struct node *rear;

```

```
void insert();
void delete();
void display();
void insert()
{
    struct node *temp;

    int x;

    temp=(struct node *)malloc(sizeof(struct node));
    if(temp == NULL)
    {
        printf("\n overflow");
    }
    else
    {
        printf("\nEnter value :");
        scanf("%d",&x);
        temp->data = x;
        if(front == NULL)
        {
            front = temp;
            rear = temp;
            front->next = NULL; rear->next = NULL;
        }
        else
        {
            rear->next = temp;
            rear = temp;
        }
    }
}
```

```
rear->next = NULL;

}

}

}

void delete()
{
    struct node *temp;
    if(front == NULL)
    {
        printf("\nUnderflow"); }
    else
    {
        temp = front;
        front = front->next;
    }
}

void display()
{
    struct node *temp;
    temp = front;
    if(front == NULL)
    {
        printf("\nQueue is empty"); }
    else
    {
        printf("\nPrinting value"); while(temp!=NULL)
    {
```



```
printf("\n%d",temp->data); temp = temp->next; }  
}  
}  
void main()  
{  
    int choice;  
    while(1)  
    {  
        printf("\n1.Insert\n2.Delete\n3.Display\n4.Exit"); printf("\nEnter your choice :");  
        scanf("%d",&choice);  
        switch(choice)  
        {  
            case 1:  
                insert();  
                break;  
  
            case 2:  
                delete();  
                break;  
  
            case 3:  
                display();  
                break;  
  
            case 4:  
                exit (0);  
        }  
    }
```

```
}  
  
}
```

OUTPUT:

```
1.Insert  
2.Delete  
3.Display  
4.Exit  
Enter your choice :1  
  
Enter value :10  
  
1.Insert  
2.Delete  
3.Display  
4.Exit  
Enter your choice :2  
  
1.Insert  
2.Delete  
3.Display  
4.Exit  
Enter your choice :3  
  
Queue is empty  
1.Insert  
2.Delete  
3.Display  
4.Exit  
Enter your choice :[]
```