5

USN: 1BM19CS016 Name: AMIT.R

// Linked Stack Code

lab8:

```
#include <stdio.h>
#include <conio.h>
void push (int);
void pop();
void display();
struct node
{
```

```
        int data;
        struct node *next;
};
struct node *top = NULL;
    void    main()
{
        int choice, ch;
        do
        {
            printf("\n1. Push 2. Display 3. Pop 4exit\n");
            printf("\n Enter choice: ");
            scanf("%d", &choice);
            switch (choice)
            {
                case 1: printf("Enter element \n");
                        scanf("%d", &ch);
                        push(ch); break;
                case 2: display(); break;
                case 3: pop(); break;
                case 4: exit(0);
            }
        } while (1);
}
void push(int item)
{
        struct node *newnode;
        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 empty!!");
        else
        {
            printf ("element removed: %d",
            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;
        }
    }
```

```
// Linked Queue Code
#include <stdio.h>
#include <stdlib.h>
struct node
{
    int data;
```

```
    struct node *next;
};
void insert (int);
void display ();
void del ();
struct node *rear = NULL, *front = NULL;
void main()
{
    int choice, c;
    char ch = 'y';
    do
    {
    printf ("\n Queue implementation using linklist\n");
    printf ("\n 1. Create 2. Display 3. Delete
                                4. Exit \n");
    printf ("\n Enter choice \n");
    scanf ("%d", &choice);
    switch (choice)
    {       case 1: printf ("enter element \n");
                    scanf ("%d", &c);
                    insert(c); break;
            case 2: display(); break;
            case 3: del(); break;
            case 4: exit(0);
    }
    } while (1);
}
void insert (int item)
{
    struct node *newnode;
    newnode = (struct node *) malloc (sizeof (struct node));
    newnode -> data = item;
```

```
        newnode -> next = NULL;
        if (rear == NULL)
        {
            rear = newnode;
            front = newnode;
        }
        else
        {
            rear -> next = newnode;
            rear = newnode;
        }
}
void dd()
{
    if (front == NULL)
    printf ("Queue empty !\n");
    else
    {
        printf ("Deleted ele is %d", front -> data);
        if (front == rear)
        {
            printf ("Queue is empty \n");
            front = rear = NULL;
        }
        else
            front = front -> next;
}
void display()
{
    struct node *temp;
    if (front == NULL)
    {
        printf ("Queue empty \n");
        return;
    }
```

```
temp = fcont;
while (temp != NULL)
{
    printf ("%d ", temp -> data);
    temp = temp -> next;
}
}
```