

INDEX

- 1. Introduction to data structures**
- 2. Array Implementation Of Stack**
- 3. Application Of Stack – Conversion Of Infix To Postfix**
- 4. Implementation Of Linear Queue Using Arrays**
- 5. Array Implementation Of Circular Queue**
- 6. Implementation of Singly Linked List**
- 7. Implementation of Stack using Linked List**
- 8.**

Ex. No.:7

Date :

Implementation of Stack using Linked List

Aim: To write a C-program to implement Stack using Linked List.

Program

```
/*This program demonstrate the linked implementation of stacks*/
#include <process.h>
#include <stdio.h>
#include <stdlib.h>
#include <conio.h>

struct stack
{
    int no;
    struct stack *next;
} *top = NULL;

typedef struct stack st;
void push();
int pop();
void display();
int main()
{
    char ch;
    int choice, item;
    int flag = 1;
    do
    {
        printf("\n\n Enter your choice");
        printf(" \n\n\t 1: Push the elements");
        printf(" \n\n\t 2: Pop the elements");
        printf(" \n\n\t 3: To display the element");
        printf(" \n\n\t 4: Exit");
        printf("\n\n\n Enter of your choice:\t");
        scanf("%d",&choice);
        //clrscr();
        switch(choice)
        {
            case 1:
                push();
                break;
            case 2:
                item = pop();
                if(item != -1)
                    printf("poped item is %d", item);
                break;
            case 3:
                display();
                break;
            case 4:
                exit(0);
                break;
        }
    } while(ch != 'q');
    //clrscr();
}
```

```
        display();
        break;
    case 4:
        flag=0;
        break;
    default:
        printf("\n Invalid Choice");
    }
    }while(flag);
    getch();
}
```

```
void push()
{
    st *p;
    p = (st *) malloc (sizeof(st));
    printf("\n Enter the number");
    scanf("%d", &p->no);
    p->next = top;
    top = p;
}
```

```
int pop()
{
    st *p;
    p = top;
    if(top == NULL){
        printf("stack is already empty");
        return -1;
    }
    else
    {
        top = top -> next;
        return (p -> no);
        free(p);
    }
}
```

```
void display()
{
    st *p;
    p = top;
    while (p != NULL)
    {
        printf("\nno = %d", p -> no);
        p = p -> next;
    }
    if(top == NULL)
        printf("Stack is empty");
}
```