

S.No: 11

Exp. Name: **Write a C program to implement different Operations on Stack using Linked Lists**

Date: 2022-08-03

Aim:Write a program to implement `stack` using **linked lists**.

Sample Input and Output:

```
1.Push 2.Pop 3.Display 4.Is Empty 5.Peek 6.Exit
Enter your option : 1
Enter element : 33
Successfully pushed.
1.Push 2.Pop 3.Display 4.Is Empty 5.Peek 6.Exit
Enter your option : 1
Enter element : 22
Successfully pushed.
1.Push 2.Pop 3.Display 4.Is Empty 5.Peek 6.Exit
Enter your option : 1
Enter element : 55
Successfully pushed.
1.Push 2.Pop 3.Display 4.Is Empty 5.Peek 6.Exit
Enter your option : 1
Enter element : 66
Successfully pushed.
1.Push 2.Pop 3.Display 4.Is Empty 5.Peek 6.Exit
Enter your option : 3
Elements of the stack are : 66 55 22 33
1.Push 2.Pop 3.Display 4.Is Empty 5.Peek 6.Exit
Enter your option : 2
Popped value = 66
1.Push 2.Pop 3.Display 4.Is Empty 5.Peek 6.Exit
Enter your option : 2
Popped value = 55
1.Push 2.Pop 3.Display 4.Is Empty 5.Peek 6.Exit
Enter your option : 3
Elements of the stack are : 22 33
1.Push 2.Pop 3.Display 4.Is Empty 5.Peek 6.Exit
Enter your option : 5
Peek value = 22
1.Push 2.Pop 3.Display 4.Is Empty 5.Peek 6.Exit
Enter your option : 4
Stack is not empty.
1.Push 2.Pop 3.Display 4.Is Empty 5.Peek 6.Exit
Enter your option : 6
```

Source Code:

StackUsingLL.c

```
#include <stdio.h>
#include <stdlib.h>
#include "StackOperationsLL.c"

int main() {
    int op, x;
```

Page No:

ID: 219X1A04E7

2021-2025-ECE-B

G Pulla Reddy Engineering College (Autonomous)

```
while(1) {
    printf("1.Push 2.Pop 3.Display 4.Is Empty 5.Peek 6.Exit\n");
    printf("Enter your option : ");
    scanf("%d", &op);
    switch(op) {
        case 1:
            printf("Enter element : ");
            scanf("%d", &x);
            push(x);
            break;
        case 2:
            pop();
            break;
        case 3:
            display();
            break;
        case 4:
            isEmpty();
            break;
        case 5:
            peek();
            break;
        case 6:
            exit(0);
    }
}
```

StackOperationsLL.c

```
struct stack {
    int data;
    struct stack *next;
};
typedef struct stack *stk;
stk top=NULL;
void push(int ele)
{
    stk temp;
    temp=(stk)malloc(sizeof(struct stack));
    temp->data=ele;
    if(temp==NULL)
        printf("Stack is overflow.\n");
    else
    {
        temp->next=top;
        top=temp;
        printf("Successfully pushed.\n");
    }
}
void pop()
{
    stk temp;
    if(top==NULL)
        printf("Stack is underflow.\n");
```

```

    else
    {
        temp=top;
        printf("Popped value = %d\n",top->data);
        top=top->next;
        free(temp);
    }
}
void peek()
{
    if(top==NULL)
        printf("Stack is underflow.\n");
    else
        printf("Peek value = %d\n",top->data);
}
void isEmpty()
{
    printf("Stack is ");
    if(top==NULL)
        printf("empty.\n");
    else
        printf("not empty.\n");
}
void display()
{
    stk temp=top;
    if(top==NULL)
        printf("Stack is empty.\n");
    else {
        printf("Elements of the stack are : ");
        while(temp!=NULL)
        {
            printf("%d ",temp->data);
            temp=temp->next;
        }
        printf("\n"); }
}

```

Execution Results - All test cases have succeeded!

Test Case - 1
User Output
1.Push 2.Pop 3.Display 4.Is Empty 5.Peek 6.Exit 1
Enter your option : 1
Enter element : 33
Successfully pushed. 1
1.Push 2.Pop 3.Display 4.Is Empty 5.Peek 6.Exit 1
Enter your option : 1
Enter element : 22
Successfully pushed. 1
1.Push 2.Pop 3.Display 4.Is Empty 5.Peek 6.Exit 1
Enter your option : 1

Test Case - 1

Enter element : 55
 Successfully pushed. 1
 1.Push 2.Pop 3.Display 4.Is Empty 5.Peek 6.Exit 1
 Enter your option : 1
 Enter element : 66
 Successfully pushed. 3
 1.Push 2.Pop 3.Display 4.Is Empty 5.Peek 6.Exit 3
 Enter your option : 3
 Elements of the stack are : 66 55 22 33 2
 1.Push 2.Pop 3.Display 4.Is Empty 5.Peek 6.Exit 2
 Enter your option : 2
 Popped value = 66 2
 1.Push 2.Pop 3.Display 4.Is Empty 5.Peek 6.Exit 2
 Enter your option : 2
 Popped value = 55 3
 1.Push 2.Pop 3.Display 4.Is Empty 5.Peek 6.Exit 3
 Enter your option : 3
 Elements of the stack are : 22 33 5
 1.Push 2.Pop 3.Display 4.Is Empty 5.Peek 6.Exit 5
 Enter your option : 5
 Peek value = 22 4
 1.Push 2.Pop 3.Display 4.Is Empty 5.Peek 6.Exit 4
 Enter your option : 4
 Stack is not empty. 6
 1.Push 2.Pop 3.Display 4.Is Empty 5.Peek 6.Exit 6
 Enter your option : 6

Test Case - 2**User Output**

1.Push 2.Pop 3.Display 4.Is Empty 5.Peek 6.Exit 2
 Enter your option : 2
 Stack is underflow. 3
 1.Push 2.Pop 3.Display 4.Is Empty 5.Peek 6.Exit 3
 Enter your option : 3
 Stack is empty. 5
 1.Push 2.Pop 3.Display 4.Is Empty 5.Peek 6.Exit 5
 Enter your option : 5
 Stack is underflow. 4
 1.Push 2.Pop 3.Display 4.Is Empty 5.Peek 6.Exit 4
 Enter your option : 4
 Stack is empty. 1
 1.Push 2.Pop 3.Display 4.Is Empty 5.Peek 6.Exit 1
 Enter your option : 1
 Enter element : 23
 Successfully pushed. 1
 1.Push 2.Pop 3.Display 4.Is Empty 5.Peek 6.Exit 1
 Enter your option : 1
 Enter element : 24
 Successfully pushed. 3

Test Case - 2
1.Push 2.Pop 3.Display 4.Is Empty 5.Peek 6.Exit 3
Enter your option : 3
Elements of the stack are : 24 23 5
1.Push 2.Pop 3.Display 4.Is Empty 5.Peek 6.Exit 5
Enter your option : 5
Peek value = 24 2
1.Push 2.Pop 3.Display 4.Is Empty 5.Peek 6.Exit 2
Enter your option : 2
Popped value = 24 2
1.Push 2.Pop 3.Display 4.Is Empty 5.Peek 6.Exit 2
Enter your option : 2
Popped value = 23 2
1.Push 2.Pop 3.Display 4.Is Empty 5.Peek 6.Exit 2
Enter your option : 2
Stack is underflow. 4
1.Push 2.Pop 3.Display 4.Is Empty 5.Peek 6.Exit 4
Enter your option : 4
Stack is empty. 6
1.Push 2.Pop 3.Display 4.Is Empty 5.Peek 6.Exit 6
Enter your option : 6