

**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;
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 node
{
    int info;
    struct node *next;
};
struct node *FIRST=NULL;
struct node *top=NULL;
struct node *newNode(int);

struct node *newNode(int x)
{
    struct node *temp=(struct node*)malloc(sizeof(struct node));
    if(temp==NULL)
    {
        printf("{Unable to allocate memory .\nNode not created.\n}");
        exit(0);
    }
    else
    {
        temp->info=x;
        temp->next=NULL;
        return temp;
    }
}

```

```
void push(int x)
{
    struct node *temp,*ptr,*ptr1;
    temp=newNode(x);
    if(FIRST==NULL || top==NULL)
    {
        FIRST=temp;
        top=temp;
        printf("Successfully pushed.\n");
    }
    else
    {
        ptr=FIRST;
        while(ptr->next!=NULL)
        {
            ptr=ptr->next;
        }
        ptr->next=temp;
        top=temp;
        printf("Successfully pushed.\n");
    }
}

void pop()
{
    struct node *ptr;
    if(top==NULL || FIRST==NULL)
    {
        printf("Stack is underflow.\n");
    }
    else if(FIRST==top)
    {
        printf("Popped value = %d\n",top->info);
        free(top);
        top=NULL;
        FIRST=NULL;
    }
    else
    {
        ptr=FIRST;
        while(ptr->next!=top)
        {
            ptr=ptr->next;
        }
        printf("Popped value = %d\n",top->info);
        top=ptr;
        ptr=ptr->next;
        free(ptr);
    }
}

void display()
{
    struct node *ptr;
    if(FIRST==NULL || top==NULL)
    {
        printf("Stack is empty.\n");
    }
}
```

```

else
{
    int a[10];
    ptr=FIRST;
    int i=0;
    while(ptr!=top)
    {
        a[i++]=ptr->info;
        ptr=ptr->next;
    }
    a[i++]=ptr->info;
    int j=i-1;
    printf("Elements of the stack are : ");
    for(int k=j;k>=0;k--)
    {
        printf("%d ",a[k]);
    }
    printf("\n");
}
}
void isEmpty()
{
    if(FIRST==NULL || top==NULL)
    {
        printf("Stack is empty.\n");
    }
    else
    {
        printf("Stack is not empty.\n");
    }
}
void peek()
{
    if(FIRST==NULL)
    {
        printf("Stack is underflow.\n");
    }
    else
    {
        printf("Peek value = %d\n",top->info);
    }
}
}

```

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
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
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