# Implementation of Stack Data Structure using Linked List

```
#include<stdio.h>
#include<stdlib.h>
int count=0;

struct node                                     //Define a node of Stack
{
        int no;
        struct node *next;
}*new, *first=NULL, *ptr;

void get_element()                              //Create a new node & store data
{
    new = (struct node*) malloc(sizeof(struct node));
    printf("New Address: %p\n", new);
    printf("Enter data to insert: ");
    scanf("%d", &new->no);                      //Get data
        new->next=NULL;
}

void insert_begin()                             //Insert a Node into Stack (PUSH) LIFO/FILO
{
        get_element();                          //Create a new node
        count = count+1;
        if(first==NULL) first=new;              //Head Node
        else
        {
                new->next=first;
                first=new;                      //Change Head node after every insertion
        }
        printf("Element inserted in Stack\n");
}

void create_list()                              //Create or append 'n' nodes
{
        int n;
        printf("Enter the number of elements to be inserted: ");
        scanf("%d", &n);
        for(int i=1; i<=n; i++)
        {
                insert_begin();                 //Insert Node into Stack
                printf("\nElement-%d inserted\n", i);
        }
}

void insert()                                   //Menu Driven Insert Procedure
{
        int choice;
        L1: printf("\nEnter 1(Insert One), 2(Insert Multiple): ");
        scanf("%d", &choice);
        switch(choice)
        {
                case 1: insert_begin(); break;
                case 2: create_list(); break;
                default: printf("Wrong choice\n");
        }
}

void display()                  //Traverse the stack & fetch all its elements in order
{
        if(first==NULL) printf("Stack is empty\n");                           //Stack underflow
        else
        {
                printf("\nNo. of elements in Stack: %d\n", count);
                for(ptr=first; ptr!=NULL; ptr=ptr->next)                  //Traverse
                        printf("Block Address:%p, Data: %d, Next: %p\n", ptr, ptr->no, ptr->next);        //Fetch data
        }
}

void delete_begin()             //Delete a Node from Stack (POP) LIFO/FILO
{
        if(first!=NULL)
        {
                if(first->next==NULL) first=NULL;       //When the stack has one node
                else
                {
                        ptr=first->next;
                        first=ptr;                      //Replace the Head Node
                }
                count = count-1;
                printf("Element is Deleted from Stack\n");
        }
        else printf("List is Empty\n");
}

int main()
{
        int choice;
        L1: printf("\nEnter 1(Insert), 2(Delete), 3(Display), 4(Exit): ");
        scanf("%d", &choice);
        switch(choice)
        {
                case 1: insert(); goto L1;
                case 2: delete_begin(); goto L1;
                case 3: display(); goto L1;
                case 4: break;
                default: printf("Wrong Choice\n"); goto L1;
        }
        return 0;
}
```