Implement a Stack in C Programming

```
#include <stdio.h>

#include <stdlib.h>

#define SIZE 4

int top = -1, inp_array[SIZE];
void push();
void pop();
void show();

int main()
{
	int choice;

	while (1)
	{
		printf("\nPerform operations on the stack:");
		printf("\n1.Push the element\n2.Pop the element\n3.Show\n4.End");
		printf("\n\nEnter the choice: ");
		scanf("%d", &choice);

		switch (choice)
```

```
            {
            case 1:
                push();
                break;
            case 2:
                pop();
                break;
            case 3:
                show();
                break;
            case 4:
                exit(0);

            default:
                printf("\nInvalid choice!!");
            }
        }
}

void push()
{
        int x;

        if (top == SIZE - 1)
        {
            printf("\nOverflow!!");
```

```
        }
        else
        {
                printf("\nEnter the element to be added onto the stack: ");
                scanf("%d", &x);
                top = top + 1;
                inp_array[top] = x;
        }
}

void pop()
{
        if (top == -1)
        {
                printf("\nUnderflow!!");
        }
        else
        {
                printf("\nPopped element: %d", inp_array[top]);
                top = top - 1;
        }
}

void show()
{
        if (top == -1)
```

```
    {
        printf("\nUnderflow!!");
    }
    else
    {
        printf("\nElements present in the stack: \n");
        for (int i = top; i >= 0; --i)
            printf("%d\n", inp_array[i]);
    }
}
```

OUTPUT

Perform operations on the stack:

1.Push the element

2.Pop the element

3.Show

4.End

Enter the choice: 1

Enter the element to be inserted onto the stack: 10

STACK IMPLEMENTATION USING LINKED LIST

```
#include <stdio.h>
#include <stdlib.h>
struct node {
```

```
        int info;
        struct node *ptr;
}*top,*top1,*temp;

int count = 0;
// Push() operation on a     stack
void push(int data) {
          if (top == NULL)
        {
                top =(struct node *)malloc(1*sizeof(struct node));
                top->ptr = NULL;
                top->info = data;
        }
        else
        {
                temp =(struct node *)malloc(1*sizeof(struct node));
                temp->ptr = top;
                temp->info = data;
                top = temp;
        }
        count++;
        printf("Node is Inserted\n\n");
}

int pop() {
          top1 = top;
```

```
        if (top1 == NULL)
        {
                printf("\nStack Underflow\n");
                return -1;
        }
        else
                top1 = top1->ptr;
        int popped = top->info;
        free(top);
        top = top1;
        count--;
        return popped;
}

void display() {
        // Display the elements of the stack
        top1 = top;

        if (top1 == NULL)
        {
                printf("\nStack Underflow\n");
                return;
        }

        printf("The stack is \n");
```

```
    while (top1 != NULL)
    {
        printf("%d--->", top1->info);
        top1 = top1->ptr;
    }
    printf("NULL\n\n");

}

int main() {
    int choice, value;
    printf("\nImplementation of Stack using Linked List\n");
    while (1) {
        printf("\n1. Push\n2. Pop\n3. Display\n4. Exit\n");
        printf("\nEnter your choice : ");
        scanf("%d", &choice);
        switch (choice) {
        case 1:
            printf("\nEnter the value to insert: ");
            scanf("%d", &value);
            push(value);
            break;
        case 2:
            printf("Popped element is :%d\n", pop());
            break;
        case 3:
```

```
                display();
                break;
        case 4:
                exit(0);
                break;
        default:
                printf("\nWrong Choice\n");
        }
    }
}
```

OUTPUT

Implementation of Stack using Linked List

1. Push

2. Pop

3. Display

4. Exit

Enter your choice : 1

Enter the value to insert: 2 3 5

Node is Inserted