

```

/*Program for demonstrating implementation of stacks using linked list*/

#include <stdio.h>

#include <stdlib.h>

struct stack /*Declaring the structure for stack elements*/
{
    int element;
    struct stack *next; /*Stack element pointing to another stack element*/
} *top;

void push(int); /*Declaring a function prototype for inserting an element      p into the stack*/

int pop(); /*Declaring a function prototype for removing an element from      e
the stack*/

void display(); /*Declaring a function prototype for displaying the elements of a stack*/

void main()
{
    int num1, num2, choice;

    while(1)
    {

        /*Creating an interactive interface for performing stack operations*/
        printf("Select a choice from the following:");
        printf("\n[1] Push an element into the stack");
        printf("\n[2] Pop out an element from the stack");
        printf("\n[3] Display the stack elements");
        printf("\n[4] Exit\n");
        printf("\n\tYour choice: ");
        scanf("%d",&choice);
    }
}

```

```
switch(choice)
{
case 1:
{
printf("\n\tEnter the element to be pushed into the stack: ");
scanf("%d",&num1);

push(num1); /*Inserting an element*/
break;
}

case 2:
{
num2=pop(); /*Removing an element*/
printf("\n\t%d element popped out of the stack\n\t",num2);
break;
}

case 3:
{
display(); /*Displaying stack elements*/

break;
}

case 4:
exit(1);
break;

default:
printf("\nInvalid choice!\n"); break;
```

```
}  
}  
}
```

```
/*Push function*/ void push(int value)
```

```
{
```

```
struct stack *ptr;
```

```
ptr=(struct stack*)malloc(sizeof(struct stack)); /*Dynamically allocating memory space to store stack  
element*/
```

```
ptr->element=value; /*Assigning value to the newly allocated stack element*/
```

```
/*Updating stack pointers*/ ptr->next=top;
```

```
top=ptr;
```

```
return;
```

```
}
```

```
/*Pop function*/
```

```
int pop()
```

```
{
```

```
if(top==NULL) /*Checking whether the stack is empty*/
```

```
{
```

```
printf("\nSTACK is Empty.");
```

```
exit(1);
```

```
}
```

```
else
```

```
{
```

```
int temp=top->element; /* Retrieving the top element*/
```

```
top=top->next; /*Updating the stack pointer*/
```

```
return (temp); /*Returning the popped value*/
```

```
}
```

```

}

void display()
{
    struct stack *ptr1=NULL; ptr1=top;
    printf("\nThe various stack elements are:\n");
    while(ptr1!=NULL)
    {
        printf("%d\t",ptr1->element); /*Printing stack elements*/
        ptr1=ptr1->next;
    }
}

```

Output:

Select a choice from the following:

- [1] Push an element into the stack
- [2] Pop out an element from the stack
- [3] Display the stack elements
- [4] Exit

Your choice: 1

Enter the element to be pushed into the stack: 10

Select a choice from the following:

- [1] Push an element into the stack
- [2] Pop out an element from the stack
- [3] Display the stack elements
- [4] Exit

Your choice: 1

Enter the element to be pushed into the stack: 20

Select a choice from the following:

- [1] Push an element into the stack
- [2] Pop out an element from the stack
- [3] Display the stack elements
- [4] Exit

Your choice: 3

The various stack elements are:

20 10 Select a choice from the following:

- [1] Push an element into the stack
- [2] Pop out an element from the stack
- [3] Display the stack elements
- [4] Exit

Your choice: 1

Enter the element to be pushed into the stack: 55

Select a choice from the following:

- [1] Push an element into the stack
- [2] Pop out an element from the stack
- [3] Display the stack elements
- [4] Exit

Your choice: 3

The various stack elements are:

55 20 10 Select a choice from the following:

- [1] Push an element into the stack
- [2] Pop out an element from the stack

[3] Display the stack elements

[4] Exit