```
#include<stdlib.h>
#include<stdio.h>

typedef struct cllstack {
        int data;
        struct cllstack * ptr;
}cllstack;

cllstack * head = NULL;

cllstack * createnode() {
        cllstack * newnode = (cllstack *)malloc(sizeof(cllstack));
        printf("Enter the data element : ");
        scanf("%d" , &newnode->data);
        newnode->ptr = NULL;
        return newnode;
}

void push() {
        if(head == NULL) {
                head = createnode();
                head->ptr = head;
        }
        else {
                cllstack * temp = createnode();
                temp->ptr = head->ptr;
                head->ptr = temp;
        }
        printf("Element successfully pushed.\n");
}

void pop() {
        if(head == NULL)
                printf("Stack is empty.\n");
        else if(head->ptr == head) {
                printf("Element %d successfully popped out.\n" , head-
>data);
                free(head);
                head = NULL;
        }
        else {
                cllstack * temp = head->ptr;
                printf("Element %d successfully deleted.\n" , temp-
>data);
                temp = temp->ptr;
                head->ptr = temp;
        }
}

void display() {
        if(head == NULL)
                printf("Stack is empty.\n");
        else {
                cllstack * strt = head->ptr ;
                printf("Elements of stack are : ");
                while(strt != head) {
```

```
                        printf("%d  " , strt->data);
                        strt = strt->ptr;
                }
                printf("%d  " , strt->data);
                printf("\n");
                strt = NULL;
        }
}

void main() {
        int choice , flag = 1 ;
        printf("Enter the operation to be performed.\n");
        while(flag) {
                printf("1.Push \t 2.Pop \t 3.Display \t 4.Exit  : ");
                scanf("%d" , &choice);
                switch(choice) {
                        case 1 : push();
                                        break;

                        case 2 : pop();
                                        break;

                        case 3 : display();
                                        break;

                        case 4 : flag = 0 ;
                                    break;

                        default : printf("Invalid input.\n");
                                     break;
                }
        }
}
```