```c
#include<stdio.h>
#include<stdlib.h>
#include<malloc.h>
struct node
{
int data;
struct node * next;

};
struct node *head=NULL;
void stacksinglypush();
void stacksinglypop();
void stackdisplay();

void stacksinglypush()
{
struct node *newnode;
newnode=(struct node*)malloc(sizeof(struct node));
printf("enter data:");
scanf("%d",&newnode->data);
newnode->next=NULL;
if(head==NULL)
{
head=newnode;
}
else
{
newnode->next=head;
head=newnode;
}
}

void stacksinglypop()
{
    {
    struct node*temp;

    if(head==NULL)
    {
        printf("list is empity");
    }
    else if(head->next==NULL)
    {
        head=NULL;
        //free(temp);
        printf("\n stack deleted\n");
    }
    else
    {
        temp=head;
        head=head->next;
        free(temp);
        //printf("\n begin node deleted\n");
    }
}


}

void stackdisplay()
{
struct node *temp;
if(head==NULL)
{
printf("list is empty\n");
}
```

```c
else
{
temp=head;
printf("List elements\n");
while(temp!=NULL)
{
printf("%d\t",temp->data);
temp=temp->next;
}
}
}

int main()
{
    int op;
    while(op !=4)
    {
    int op;
    printf("\n********manu*********\n");
    printf("\n 1->stack as singly push\n");
    printf("\n 2->stack as singly pop\n");
    printf("\n 3->stack as singly display\n");
    printf("\n 4->exit\n");
    printf("\n enter your choice\n");
    scanf("%d",&op);
    switch(op)
    {
    case 1:
          stacksinglypush();
          break;
    case 2:
          stacksinglypop();
          break;
    case 3:
            stackdisplay();
            break;
    case 4:
        exit(0);
    default:
            printf("\n incorrect choice\n");
    }
    }
    return 0;
}
```