**LAB-3**

**1.Write a program to create and print linked list - insert and delete nodes at beginning and end.**

**Ans:-**

**#include <stdio.h>**

**#include <string.h>**

**#include <math.h>**

**#include <stdlib.h>**

**#define MAXSIZE 10**

**struct node{**

**int item;**

**struct node \*next;**

**}\*head,\*last,\*temp,\*temp2;**

**int count=0;**

**void insert\_at\_beginning(struct node \*\*head,int value)**

**{**

**if(count>MAXSIZE-1)**

**{**

**printf("List can not accommodate more than %d elements",MAXSIZE);**

**exit(0);**

**}**

**temp=(struct node \*)malloc(sizeof(struct node));**

**temp->item=value;**

**temp->next=\*head;**

**\*head=temp;**

**count++;**

**}**

**void insert\_at\_end(struct node \*\*head,int value)**

**{**

**if(count> MAXSIZE-1)**

**{**

**printf("List can not accommodate more than %d elements",MAXSIZE);**

**exit(0);**

**}**

**temp2=\*head;**

**temp=(struct node \*)malloc(sizeof(struct node));**

**temp->item=value;**

**temp->next=NULL;**

**if(\*head!=NULL)**

**{**

**while(temp2->next!=NULL)**

**{**

**temp2=temp2->next;**

**}**

**temp2->next=temp;**

**}**

**else**

**{**

**\*head=temp;**

**}**

**count++;**

**}**

**void delete\_from\_beginning(struct node \*\*head)**

**{**

**if(count==0)**

**{**

**puts("INVALID SEQUENCE OF COMMANDS");**

**exit(0);**

**}**

**if(\*head!=NULL)**

**{**

**temp2=(\*head)->next;**

**\*head=temp2;**

**}**

**count--;**

**}**

**void delete\_from\_end(struct node \*\*head)**

**{**

**if(count==0)**

**{**

**puts("INVALID SEQUENCE OF COMMANDS");**

**exit(0);**

**}**

**temp=\*head;**

**count--;**

**if((\*head)->next==NULL)**

**{**

**\*head=NULL;**

**return;**

**}**

**while(temp->next!=NULL)**

**{**

**temp2=(temp->next)->next;**

**if(temp2==NULL)**

**{**

**temp->next=NULL;**

**break;**

**}**

**temp=temp->next;**

**}**

**}**

**void print\_list(struct node \*\*head)**

**{**

**if(count==0)**

**{**

**puts("List is Empty");**

**exit(0);**

**}**

**temp=\*head;**

**while(1)**

**{**

**if(temp==NULL)**

**break;**

**printf("%d ",temp->item);**

**temp=temp->next;**

**}**

**}**

**int main() {**

**int code=-1;**

**while(code!=4)**

**{**

**scanf("%d",&code);**

**if(code==0)**

**{**

**int data;**

**scanf("%d",&data);**

**if(head==NULL)**

**{**

**last=head;**

**}**

**insert\_at\_beginning(&head,data);**

**}**

**if(code==1)**

**{**

**int data;**

**scanf("%d",&data);**

**insert\_at\_end(&head,data);**

**}**

**if(code==2)**

**{**

**delete\_from\_beginning(&head);**

**}**

**if(code==3)**

**{**

**delete\_from\_end(&head);**

**}**

**if(code==4)**

**{**

**print\_list( &head);**

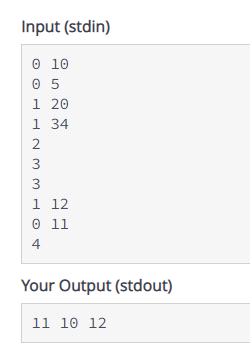
**}**

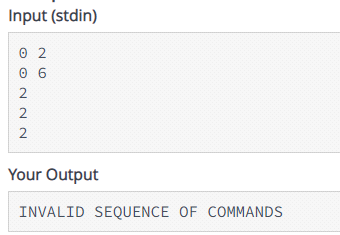
**}**

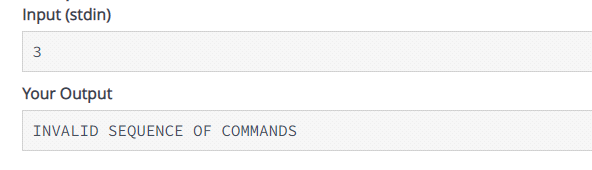
**return 0;**

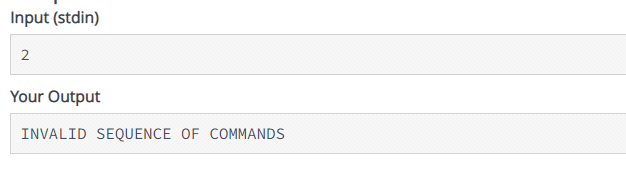
**}**

**test cases::**

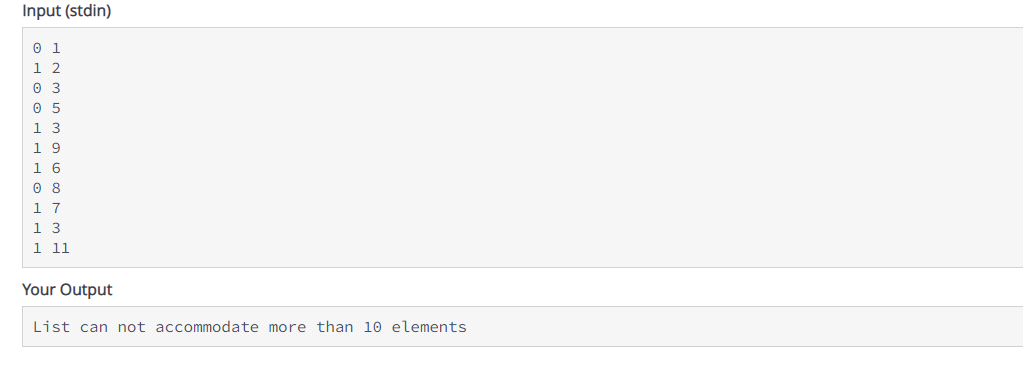












**2.Write code for insert\_at\_end function and use it to create linked list of input numbers. head\_addr in each function argument is pointer to head; and head is a pointer to first node in linked list. value in insert function represents value to be inserted at the end of linked list.**

Ans:-

**#include <stdio.h>**

**#include <string.h>**

**#include <math.h>**

**#include <stdlib.h>**

**#define MAXSIZE 10**

**struct node{**

**int item;**

**struct node \*next;**

**}\*head,\*last,\*temp,\*temp2;**

**int count=0;**

**void insert\_at\_end(struct node \*\*head\_add,int value)**

**{**

**if(count>MAXSIZE-1)**

**{**

**printf("List can not accommodate more than %d elements",MAXSIZE);**

**exit(0);**

**}**

**temp2=\*head\_add;**

**temp=(struct node \*)malloc(sizeof(struct node));**

**temp->item=value;**

**temp->next=NULL;**

**if(\*head\_add!=NULL)**

**{**

**while(temp2->next!=NULL)**

**{**

**temp2=temp2->next;**

**}**

**temp2->next=temp;**

**}**

**else**

**{**

**\*head\_add=temp;**

**}**

**count++;**

**}**

**void print\_list(struct node \*\*head\_add)**

**{**

**temp=\*head\_add;**

**while(1)**

**{**

**if(temp==NULL)**

**break;**

**printf("%d ",temp->item);**

**temp=temp->next;**

**}**

**printf("\n");**

**}**

**void print\_reverse\_list(struct node \*\*head\_add)**

**{**

**while(\*head\_add!=NULL&&(\*head\_add)->next!=NULL)**

**{**

**temp=\*head\_add;**

**while((temp->next)->next!=NULL)**

**{**

**temp=temp->next;**

**}**

**printf("%d ",temp->next->item);**

**temp->next=NULL;**

**}**

**printf("%d ",(\*head\_add)->item);**

**}**

**int main() {**

**int data,i;**

**while(1)**

**{**

**i=scanf("%d",&data);**

**if(i==-1)**

**break;**

**insert\_at\_end(&head,data);**

**}**

**if(count==0)**

**{**

**printf("List is Empty");**

**return 0;**

**}**

**else{**

**print\_list(&head);**

**print\_reverse\_list(&head);**

**}**

**return 0;**

**}**

**test cases::**

****

****

****

**3.Write a program to create linked list - insert and delete nodes from given index in linked list.**

**Ans:-**

**#include <stdio.h>**

**#include <string.h>**

**#include <math.h>**

**#include <stdlib.h>**

**#define MAXSIZE 10**

**struct node{**

**int item;**

**struct node \*next;**

**}\*head,\*last,\*temp,\*temp2;**

**int count=0;**

**void insert(struct node \*\* head,int ind,int value)**

**{**

**if(ind>count || count>MAXSIZE-1)**

**{**

**puts("INVALID SEQUENCE OF COMMANDS");**

**exit(0);**

**}**

**temp2=(struct node \*)malloc(sizeof(struct node));**

**temp2->item=value;**

**if( ind==0 &&count==0)**

**{**

**temp2->next=NULL;**

**\*head=temp2;**

**}**

**else if(ind==0)**

**{**

**temp2->next=\*head;**

**\*head=temp2;**

**}**

**else**

**{**

**int i=0;**

**temp=\*head;**

**while(1)**

**{**

**i++;**

**if(temp->next==NULL)**

**break;**

**if(i==ind)**

**break;**

**temp=temp->next;**

**}**

**if(temp->next!=NULL)**

**temp2->next=temp->next;**

**else**

**temp2->next=NULL;**

**temp->next=temp2;**

**}**

**count++;**

**}**

**void print\_list(struct node \*\*head\_add)**

**{**

**temp=\*head\_add;**

**while(1)**

**{**

**if(temp==NULL)**

**break;**

**printf("%d ",temp->item);**

**temp=temp->next;**

**}**

**printf("\n");**

**}**

**void delete\_(struct node \*\*head, int ind)**

**{**

**if(ind>count-1)**

**{**

**puts("INVALID SEQUENCE OF COMMANDS");**

**exit(0);**

**}**

**if(\*head!=NULL)**

**{**

**if(ind==0)**

**{**

**temp=(\*head)->next;**

**\*head=temp;**

**}**

**else**

**{**

**if((\*head)->next!=NULL)**

**{**

**temp=\*head;**

**while(ind>0)**

**{**

**ind--;**

**if(ind<=0)**

**break;**

**temp=temp->next;**

**}**

**}**

**temp->next=(temp->next)->next;**

**}**

**}**

**else**

**{**

**\*head=NULL;**

**}**

**count--;**

**}**

**int main() {**

**int code;**

**while(code!=3)**

**{**

**scanf("%d",&code);**

**if(code==1)**

**{**

**int ind,data;**

**scanf("%d%d",&ind,&data);**

**insert(&head,ind,data);**

**}**

**if(code==2)**

**{**

**int ind;**

**scanf("%d",&ind);**

**delete\_(&head,ind);**

**}**

**if(code==3)**

**{**

**print\_list(&head);**

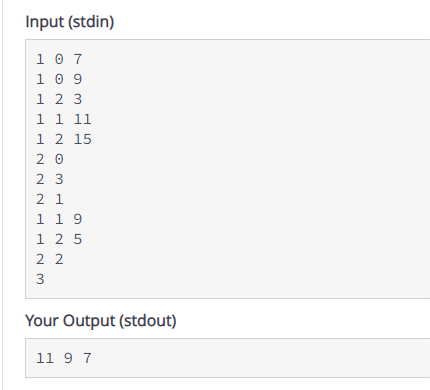
**}**

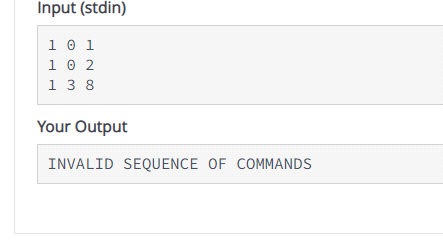
**}**

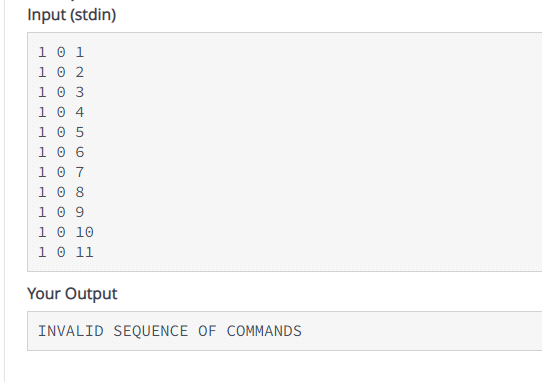
**return 0;**

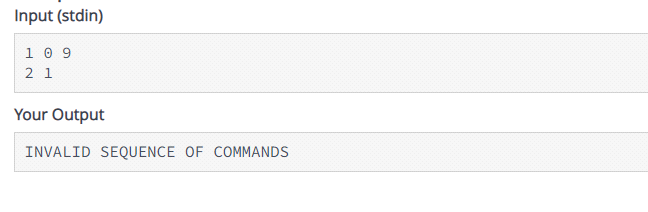
**}**

**test cases::**

****

****

****

****