LAB 4

1.Linked list Implementation

#include <stdlib.h>

#include <stdio.h>

void push();

void append();

void display();

void insert\_at\_pos();

struct node

{

int data;

struct node \*next;

};

struct node \*head=NULL;

void main()

{

printf("1.Insert from beginning\n2.Insert at end\n3.Insert at particular position\n4.Display\n5.Exit\n");

int ch;

while(ch!=6)

{

printf("Enter choice:");

scanf("%d",&ch);

switch(ch)

{

case 1:

push();

break;

case 2:

append();

break;

case 3:

insert\_at\_pos();

break;

case 4:

display();

break;

case 5:exit(0);

default: printf("Invalid choise");

break;

}

}

}

void push()

{

int data;

struct node \*new\_node;

new\_node=(struct node\*)malloc(sizeof(struct node));

printf("Enter the data to be inserted\n");

scanf("%d",&data);

new\_node->data=data;

new\_node->next=head;

head = new\_node;

}

void append()

{

int data;

struct node \*last=head;

struct node \*new\_node;

new\_node=(struct node\*)malloc(sizeof(struct node));

printf("Enter the data\n");

scanf("%d",&data);

new\_node -> data =data;

new\_node->next=NULL;

if(head==NULL)

{

head=new\_node;

return;

}

while(last->next!=NULL)

{

last=last->next;

}

last-> next =new\_node;

}

void insert\_at\_pos()

{

int data;

int pos;

struct node \*temp=head;

struct node \*new\_node;

new\_node = (struct node\*) malloc(sizeof(struct node));

printf("Enter the data to be inserted\n");

scanf("%d",&data);

new\_node->data=data;

printf("enter the position\n");

scanf("%d",&pos);

new\_node -> next = NULL;

if(pos==0)

{

new\_node->next=head;

head = new\_node;

}

else

{

for(int i=1 ; i<pos-1 ; i++)

{

temp=temp->next;

}

struct node \*temp1=temp->next;

new\_node->next= temp1;

temp->next = new\_node;

}

}

void display()

{

struct node \*p= head;

printf("List:\n");

while(p != NULL)

{

printf("%d ->",p->data);

p=p->next;

}

printf("NULL\n");

}

OUTPUT:

