#include<stdio.h>

#include<stdlib.h>

#include<malloc.h>

void create();

void display();

void insert\_head();

void insert\_last();

void insert\_val();

struct Node

{

int data;

struct Node \*link;

};

typedef struct Node node;

node \*start=NULL;

int main()

{

int ch;

while(1)

{

printf("\n1.Create\n2.Display \n3.Insert Head \n4.Insert Last\n5.Insert val\n6.Exit");

printf("\nEnter your choice:\n");

scanf("%d",&ch);

switch(ch)

{

case 1:

create();

break;

case 2:

display();

break;

case 3:

insert\_head();

break;

case 4:

insert\_last();

break;

case 5: insert\_val();

break;

case 6:

exit(1);

default:

printf("Invalid choice\n");

}

}

return 0;

}

void create()

{

int c;

node \*neww,\*curr;

start=(node \*) malloc(sizeof(node));

curr=start;

printf("Enter element\n");

scanf("%d",&start->data);

while(1)

{

printf("Do you want to add another element(1/0)\n");

scanf("%d",&c);

if(c==1)

{

neww=(node \*) malloc(sizeof(node));

printf("Enter element\n");

scanf("%d",&neww->data);

curr->link = neww;

curr=neww;

}

else

{

curr->link=NULL;

break;

}

}

}

void display()

{

node \*temp;

if(start==NULL)

{

printf("Linked list is empty\n");

return;

}

temp=start;

while(temp!=NULL)

{

printf("%d\t",temp->data);

temp = temp->link;

}

}

void insert\_head(){

node \*temp,\*mew;

mew = (node \*) malloc(sizeof(node));

temp = start;

printf("enter element value");

scanf("%d",&mew->data);

mew->link = start;

start = mew;

}

void insert\_last(){

node \*neww,\*temp;

neww = (node \*) malloc(sizeof(node));

temp = start;

printf("enter element value");

scanf("%d",&neww->data);

while(temp->link!=NULL)

{

temp = temp->link;

}

temp->link = neww;

neww->link = NULL;

}

void insert\_val(){

int pos;

node \*neww, \*temp;

neww =(node\*)malloc(sizeof(node));

printf("\nEnter element: ");

scanf("%d",&neww->data);

printf("Enter position\n");

scanf("%d",&pos);

if(pos==1)

{

neww->link=start;

start=neww;

return;

}

int i=1;

temp=start;

while(i<(pos-1) && temp!=NULL)

{

temp=temp->link;

i++;

}

if(i==(pos-1))

{

neww->link=temp->link;

temp->link=neww;

return;

}

if(temp==NULL)

{

printf("Invalid position\n");

}

}

