#include<stdlib.h>

#include<stdio.h>

#include<conio.h>

#include<malloc.h>

#include<string.h>

struct Node{

int data;

struct Node \*link;

};

char ch[5];

typedef struct Node node;

node \*new1,\*curr;

node \*start=NULL;

void create(){

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

curr=start;

printf("Enter the element\n");

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

while(1){

printf("Do you want to add another element(Y/N)\n");

scanf("%s",ch);

if(strcmp(ch,"Y")==0 ||strcmp(ch,"y")==0){

new1=(node \*)malloc(sizeof(node));

printf("Enter an element");

scanf("%d",&new1->data);

curr->link=new1;

curr=new1;

}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\_beg(){

new1=(node \*)malloc(sizeof(node));

printf("Enter element");

scanf("%d",&new1->data);

if(start==NULL){

start=new1;

new1->link=NULL;

return;

}

new1->link=start;

start=new1;

}

void insert\_end(){

node \*temp;

new1=(node \*)malloc(sizeof(node));

printf("Enter the element\n");

scanf("%d",&new1->data);

if(start==NULL){

start=new1;

new1->link=NULL;

return;

}

temp=start;

while(temp->link!=NULL){

temp=temp->link;

}

temp->link=new1;

new1->link=NULL;

}

void insert\_pos(){

node \*temp;

int pos;

new1=(node \*)malloc(sizeof(node));

printf("Enter the element");

scanf("%d",&new1->data);

printf("Enter the position");

scanf("%d",&pos);

if(pos==1){

new1->link=start;

start=new1;

return;

}

int i=1;

temp=start;

while(i<(pos-1)&&temp->link==NULL){

temp=temp->link;

i++;

}

if(i==(pos-1)){

new1->link=temp->link;

temp->link=new1;

return;

}

if(temp==NULL){

printf("Invalid position\n");

}

}

void main(){

int c;

printf("1. CREATE\t2. INSERT FIRST\t3. INSERT END\t4. INSERT AT POSITION\t5. DISPLAY\t6. EXIT\n");

while(1){

printf("\nEnter your choice");

scanf("%d",&c);

switch(c){

case 1:create();

break;

case 2:insert\_beg();

break;

case 3:insert\_end();

break;

case 4:insert\_pos();

break;

case 5:display();

break;

case 6:exit(0);

default:printf("Invalid choice\n");

}

}

}

