#include<stdio.h>

#include<stdlib.h>

struct node

{

int data;

struct node \*next;

};

struct node \*f;

struct node \*create(struct node \*first);

struct node \*display(struct node \*first);

struct node \*addbegin(struct node \*first);

struct node \*addend(struct node \*first);

struct node \*addmiddle(struct node \* first);

struct node \*delbegin(struct node \*first);

struct node \*delend(struct node \*first);

struct node \*delmiddle(struct node \*first);

struct node \*delbyposition(struct node \*first);

struct node \*addbyposition(struct node \*first);

int main()

{

int ch;

struct node \*first=NULL,\*temp,\*newnode;

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

while(1)

{

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

scanf("%d",&ch);

switch(ch)

{

case 1:first=create(first);

break;

case 2:display(first);

break;

case 3:first=addbegin(first);

break;

case 4:first=addmiddle(first);

break;

case 5:first=addend(first);

break;

case 6:first=delbegin(first);

break;

case 7:first=delmiddle(first);

break;

case 8:first=delend(first);

break;

case 9:first=addbyposition(first);

break;

case 10:first=delbyposition(first);

break;

case 11:exit(0);

break;

}

}

}

struct node \*create(struct node \*first)

{

int n,i;

struct node \*newnode,\*temp=first;

printf("Enter number of nodes");

scanf("%d",&n);

for(i=0;i<n;i++)

{

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

printf("Enter Data");

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

if(first==NULL)

{

first=newnode;

temp=newnode;

}

else

{

temp->next=newnode;

temp=newnode;

}

}

return first;

}

struct node \*display(struct node \*first)

{

struct node \*temp;

temp=first;

if(first==NULL)

{

printf("List Empty");

}

while(temp!=NULL)

{

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

temp=temp->next;

}

}

struct node \*addbegin(struct node \*first)

{

struct node \*newnode,\*temp;

temp=first;

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

printf("Enter Data:");

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

newnode->next=NULL;

if(first==NULL)

{

first=newnode;

temp=newnode;

}

else

{

newnode->next=first;

first=newnode;

}

return first;

}

struct node \*addmiddle(struct node \*first)

{

struct node \*newnode,\*temp;

int key;

printf("Enter Key");

scanf("%d",&key);

temp=first;

while(temp->next!=NULL)

{

if(temp->next!=NULL)

{

if(temp->data==key)

{

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

printf("Enter data to add middle");

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

newnode->next=temp->next;

temp->next=newnode;

break;

}

else

{

temp=temp->next;

}

}

return first;

}

}

struct node \*addend(struct node \*first)

{

struct node \*newnode,\*temp;

temp=first;

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

printf("Enter data to end");

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

newnode->next=NULL;

while(temp->next!=NULL)

{

temp=temp->next;

}

temp->next=newnode;

temp=newnode;

return first;

}

struct node \*delbegin(struct node \*first)

{

struct node \*temp=first;

if(first==NULL)

{

printf("deletion not possible");

}

else

{

first=first->next;

free(temp);

printf("First element deleted successfully");

}

return first;

}

struct node \*delend(struct node \*first)

{

struct node \*temp=first,\*prev;

while(temp->next!=NULL)

{

prev=temp;

temp=temp->next;

}

temp=prev->next;

prev->next=NULL;

free(temp);

printf("Last element deleted successfully");

return first;

}

struct node \*delmiddle(struct node \*first)

{

struct node \*temp=first,\*prev;

int ele;

if(first==NULL)

{

printf("Deletion not possible:");

}

else

{

printf("Enter element you want to delete:");

scanf("%d",&ele);

do

{

if(temp->data==ele)

{

prev->next=temp->next;

free(temp);

}

else

{

prev=temp;

temp=temp->next;

}

}while(temp->next!=NULL);

printf("Element deleted successfully");

}

return first;

}

struct node \*addbyposition(struct node \*first)

{

int i,p;

struct node \*new,\*temp;

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

printf("Enter data into newnode");

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

new->next=NULL;

temp=first;

printf("Enter position to Insert Newnode:");

scanf("%d",&p);

for(i=1;i<p-1;i++)

{

temp=temp->next;

if(temp==NULL)

break;

}

if(temp!=NULL)

{

new->next=temp->next;

temp->next=new;

}

else

{

printf("Invalid Loaction");

}

printf("Element Added Suuccessfully");

return first;

}

struct node \*delbyposition(struct node \*first)

{

int i,p;

struct node \*temp,\*prev;

if(first==NULL)

{

printf("List is empty:");

}

else

{

temp=first;

struct node \*addbyposition(struct node \*first)

{

int i,p;

struct node \*new,\*temp;

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

printf("Enter data into newnode");

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

new->next=NULL;

temp=first;

printf("Enter position to Insert Newnode:");

scanf("%d",&p);

for(i=1;i<p-1;i++)

{

struct node \*delbyposition(struct node \*first)

{

int i,p;

struct node \*temp,\*prev;

if(first==NULL)

{

printf("List is empty:");

}

else

{

temp=first;

prev=first;

printf("Enter position to delete node");

scanf("%d",&p);

for(i=2;i<=p;i++)

{

prev=temp;

temp=temp->next;

if(temp==NULL)

break;

}

if(temp!=NULL)

{

if(temp==first)

{

first=first->next;

}

prev->next=temp->next;

temp->next=NULL;

free(temp);

}

else

{

printf("Invalid Location");

}

}

printf("Element deleted successfully");

return first;

}