#include <stdio.h>

#include <stdlib.h>

struct Node

{

int data;

struct Node \*link;

}\*first=NULL,\*last=NULL,\*temp,\*current,\*prev;

int pos,n,item,ch,i;

typedef struct Node Node;

void insertClist();

void deleteClist();

void traverseClist();

void main()

{

int it,p;

printf("Enter the number of nodes\n");

scanf("%d",&n);

createClist(&first,&last);

while(n<=8)

{

printf("\n1.Insert\n2.Delete\n3.Traverse\n4.Exit");

printf("\nSelect your choice\n");

scanf("%d",&ch);

switch(ch)

{

case 1:

{

printf("Enter the item:\n");

scanf("%d",&it);

printf("Enter the position:\n");

scanf("%d",&p);

insertClist(&first,&last,it,p);

break;

}

case 2:

{

printf("\nEnter the node which you want to delete:\n");

scanf("%d",&p);

deleteClist(&first,&last,p);

break;

}

case 3:

{

traverseClist(first,last);

break;

}

case 4:

{

printf("\nExit");

break;

}

default:

{

printf("\nEnter a valid input");

break;

}

}

}

}

void createClist(Node \*\*first,Node \*\*last)

{

for(i=1;i<=n;i++)

{

printf("Enter the data field of node:%d\n",i);

scanf("%d",&item);

temp=(Node\*)malloc(sizeof(Node));

temp->data=item;

if(\*first==NULL)

\*first=temp;

else

(\*last)->link=temp;

\*last=temp;

}

}

void insertClist(Node \*\*first,Node \*\*last,int item,int pos)

{

temp=(Node\*)malloc(sizeof(Node));

if(temp==NULL)

{

printf("Unable to create a node\n");

return;

}

if((\*first==NULL)||(pos==1))

{

temp->data=item;

temp->link=(\*first);

if(\*last==NULL)

(\*last)=temp;

else

(\*last)->link=temp;

(\*first)=temp;

return;

}

current=(\*first);

i=2;

while(current->link!=(\*first))

{

if(i==pos)

{

temp->data=item;

temp->link=current->link;

current->link=temp;

return;

}

else

{

i++;

current=current->link;

}

}

temp->data=item;

temp->link=current->link;

current->link=temp;

(\*last)=temp;

}

void deleteClist(Node \*\*first,Node \*\*last,int pos)

{

if(\*first==NULL)

{

printf("\nList is empty");

return;

}

if(pos==1)

{

current=(\*first);

item=current->data;

if((\*first)->link==(\*first))

{(\*first)=(\*last)=NULL;}

else

{

(\*first)=(\*first)->link;

(\*last)->link=(\*first);

}

free(current);

printf("Deleted Item=%d",item);

}

current=(\*first);

i=2;

while(current->link!=(\*first))

{

if(i==pos)

{

temp=current->link;

item=temp->data;

current->link=temp->link;

free(temp);

printf("Deleted item=%d",item);

return;

}

else

current=current->link;

i++;

}

printf("\ndoesn't exist");

}

void traverseClist(Node \*first,Node \*last)

{

if(first==NULL)

{

printf("List is empty\n");

return;

}

do

{

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

first=first->link;

}

while(last->link!=first);

printf("End\n");

}



