**/\*\*\*\*circular linked list\*\*\*\*---------\*/**

/\*\* IN DELETE SPECIFIC POSITION IF YOU GIVE THE POSITION MORE THAN ELEMENTS IS NOT WORKING\*\*\*\*////



/\*

#include<stdio.h>

#include<stdlib.h>

struct node

{

int data;

struct node \*next;

};

struct node \*s1;

struct node \*insert\_begin(struct node \*start)

{

struct node \*temp,\*ptr;

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

printf("Enter the value to be inserted in the node\n");

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

temp->next=NULL;

if(start==NULL)

{

start=temp;

temp->next=start;

}

else{

ptr=start;

while(ptr->next!=start)

{

ptr=ptr->next;

}

ptr->next=temp;

temp->next=start;

start=temp;

}

return start;

}

struct node \*insert\_end(struct node \*start)

{

struct node \*temp,\*ptr;

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

printf("Enter the value to be inserted in the node\n");

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

if(start==NULL)

{

start=temp;

temp->next=start;

}

else{

ptr=start;

while(ptr->next!=start)

{

ptr=ptr->next;

}

ptr->next=temp;

temp->next=start;

}

return start;

}

struct node \*insert\_pos(struct node \*start)

{

int n;

struct node \*ptr;

struct node \*temp;

ptr=start;

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

printf("Enter the value to be added\n");

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

printf("Enter the position where the node has to be added\n");

scanf("%d",&n);

if(n==1)

{

temp->next=start;

while(ptr->next!=start)

ptr=ptr->next;

ptr->next=temp;

start=temp;

}

else{

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

{

ptr=ptr->next;

}

temp->next=ptr->next;

ptr->next=temp;

}

return start;

}

struct node \*delete\_begin(struct node \*start)

{

if(start==NULL)

{

printf("List is empty\n");

return start;

}

else if(start->next==start)

{

printf("value deleted=%d",start->data);

free(start);

start=NULL;

}

else{

struct node \*ptr=start;

printf("value deleted-%d",ptr->data);

while(ptr->next!=start)

{

ptr=ptr->next;

}

ptr->next=start->next;

free(start);

start=ptr->next;

}

return start;

}

struct node \*delete\_end(struct node \*start)

{

if(start==NULL)

{

printf("List is empty\n");

return start;

}

else if(start->next==start)

{

printf("The value deleted=%d",start->data);

free(start);

start=NULL;

}

else{

struct node \*ptr=start;

struct node \*prvptr=start;

while(ptr->next!=start)

{

prvptr=ptr;

ptr=ptr->next;

}

prvptr->next=start;

free(ptr);

}

return start;

}

struct node \*delete\_pos(struct node \*start)

{

int i,pos;

struct node\*ptr,\*temp;

if(start==NULL)

{

printf("LIst is empty\n");

return start;

}

else

{

printf("Enter the position to be deleteed\n");

scanf("%d",&pos);

if(pos==1)

{

ptr=start;

printf("the item deleted from the list is-%d",ptr->data);

while(ptr->next!=start)

{

if(ptr==NULL)

printf("value cant be deleted\n");

ptr=ptr->next;

}

ptr->next=start->next;

free(start);

start=ptr->next;

}

else

{

ptr=start;

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

{

temp=ptr;

ptr=ptr->next;

if(ptr==NULL)

{

printf("\n position not found\n");

}

}

temp->next=ptr->next;

printf("The value deleted is-%d",ptr->data);

free(ptr);

}

}

return start;

}

void display(struct node \*head)

{

struct node \*ptr;

if(head==NULL)

{

printf("List is empty\n");

return;

}

else{

ptr=head;

while(ptr->next!=head)

{

printf("%d\n",ptr->data);

ptr=ptr->next;

}

printf("%d\n",ptr->data);

}

}

void count(struct node \*start)

{

int c=0;

if(start==NULL)

{

printf("no of nodes=0\n");

return;

}

else{

struct node \*ptr;

ptr=start;

while(ptr->next!=start)

{

c++;

ptr=ptr->next;

}

c++;

printf(" the no of nodes=%d",c);

}

}

void search(struct node \*start)

{

int key;

int flag;

struct node \*ptr=start;

printf("Enter the value to be searched\n");

scanf("%d",&key);

int i=0;

if(start->data==key)

{

printf("Value found at position-%d",(i+1));

return;

}

else

{

ptr=ptr->next;

i=1;

while(ptr!=start)

{

if(ptr->data==key)

{

flag=1;

break;

}

else

flag=0;

i++;

ptr=ptr->next;

}

}

if(flag==1)

printf("element found at position=%d ",(i+1));

else

printf("element not found\n");

}

int main()

{

int choice;

int k;

while(1)

{

printf("\n 1. to add a node in the begginning\n 2. to add the node in the end\n 3.to add node at a specific position\n 4.to delete from the beginning \n 5 to delete at the end\n 6.to delete at a given position\n 7.to count the no of nodes\n 8.to search for an element\n 9. display the list\n");

scanf("%d",&choice);

switch(choice)

{

case 1:s1=insert\_begin(s1);

break;

case 2: s1=insert\_end(s1);

break;

case 3:s1=insert\_pos(s1);

break;

case 4:s1=delete\_begin(s1);

break;

case 5:s1=delete\_end(s1);

break;

case 6:s1=delete\_pos(s1);

break;

case 7:count(s1);

break;

case 8:search(s1);

break;

case 9:display(s1);

break;

}

}

}