Circular Double linked lists

#include<stdio.h>

#include<stdlib.h>

typedef struct cdlist

{

    struct cdlist \*prev;

    int data;

    struct cdlist \*next;

}node;

node \*head=NULL, \*last=NULL;

//creating a circular doubly linked list

void create()

{

    char rep='y';

    node \*ptr=NULL,\*cur=NULL;

    do

    {

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

        printf("Enter the data for the new node:");

        scanf("%d",&cur->data);

        cur->next=NULL;

        cur->prev=NULL;

        if(head==NULL)

        {

            head=cur;

            last=cur;

        }

        else

        {

            last->next=cur;

            cur->prev=last;

            last=cur;

            last->next=head;

            head->prev=last;

        }

        printf("Do u want to create another node y/n:");

        scanf(" %c",&rep);

    } while (rep=='y'||rep=='Y');

}

//Display forward

void display\_forward()

{

    node \*ptr=NULL;

    if(head==NULL)

        printf("List is empty\n");

    else

    {

        printf("List is\n");

        ptr=head;

        do

        {

            printf("%d<=>",ptr->data);

            ptr=ptr->next;

        } while (ptr!=head);

        printf("HEAD\n");

    }

}

//Display in reverse direction

void display\_rev()

{

node \*ptr=NULL;

if(head==NULL)

    printf("List is empty\n");

else

{

printf("List is\n");

ptr=last;

do

{

    printf("%d<=>",ptr->data);

    ptr=ptr->prev;

}while(ptr!=last);

printf("LAST\n");

}

}

//Insert a new node at the beg

void insert\_first()

{

    node \*cur=NULL;

    cur=(node\*)malloc(sizeof(node));

    printf("Enter data for the new node:");

    scanf("%d",&cur->data);

    cur->next=NULL;

    cur->prev=NULL;

    if(head==NULL) //no exisitng list

    {

        head=cur;

        last=cur;

    }

    else //if a list is exisisting already

    {

        cur->next=head;

        head->prev=cur;

        head=cur;

        last->next=head;

        head->prev=last;

    }

    printf("Node inserted successfully\n");

}

//Insert a new node at the end of the list

void insert\_end()

{

node \*ptr=NULL,\*cur=NULL;

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

printf("Enter data for the new node:");

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

cur->next=NULL;

cur->prev=NULL;

if(head==NULL)

{

    head=cur;

    last=cur;

}

else

{

    last->next=cur;

    cur->prev=last;

    last=cur;

    last->next=head;

    head->prev=last;

}

printf("Node inserted at the end successfully\n");

}

//Insert a new node after a particular node

void insert\_after()

{

    node \*ptr=NULL,\*cur=NULL;

    int key,flag=0;

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

    printf("Enter data for the new node:");

    scanf("%d",&cur->data);

    cur->next=NULL;

    cur->prev=NULL;

    if(head==NULL)

    {

        head=cur;

        last=cur;

    }

    else

    {

        printf("Enter the key value after which u want to insert a new node:");

        scanf("%d",&key);

        ptr=head;

        do

        {

            if(ptr->data==key)

            {

                flag=1;

                break;

            }

            else

                ptr=ptr->next;

        }while(ptr!=head);

        if(flag==0)

            printf("%d is not found\n",key);

        else if(ptr==last) //if we need to insert after the last node

             insert\_end();

        else

        {

            cur->next=ptr->next;

            ptr->next->prev=cur;

            ptr->next=cur;

            cur->prev=ptr;

        }

    }

    printf("Node inserted after %d successfully\n",key);

}

//Delete the first node from the list

void delete\_first()

{

node \*temp=NULL;

if(head==NULL)

    printf("List is empty\n");

else

{

    temp=head;

    head=head->next;

    last->next=head;

    head->prev=last;

    free(temp);

    printf("First node deleted\n");

}

}

//Delete the last node from the list

void delete\_last()

{

node \*temp=NULL;

if(head==NULL)

    printf("List is empty\n");

else if(head==last) //if only one node present

{

    free(head);

    head=NULL;

    last=NULL;

    printf("Last node deleted\n");

}

else

{

    temp=last;

    last=last->prev;

    last->next=head;

    head->prev=last;

    free(temp);

    printf("Last node deleted successfully\n");

}

}

//Delete a node after a particular node

void delete\_after()

{

node \*ptr=NULL;

node \*temp=NULL;

int key,flag=0;

if(head==NULL)

    printf("List is empty\n");

else

{

printf("Enter the data after which u want to delete:");

scanf("%d",&key);

ptr=head;

do

{

    if(ptr->data==key)

    {

        flag=1;

        break;

    }

    else

        ptr=ptr->next;

} while (ptr!=head);

if(flag==0)

    printf("Node %d is not found\n",key);

else

{

 if(ptr==last)   //delete  a node after the last node

    delete\_first();

else if(ptr->next==last) //delete a node after last but one node

    delete\_last();

else //delete a node some where in the middle of the list

{

    temp=ptr->next;

    ptr->next=ptr->next->next;

    ptr->next->next->prev=ptr;

    free(temp);

    printf("Node after %d is deleted successfully\n",key);

}

}

}

}

//searching for a node

void search()

{

    int flag=0,key;

    node \*ptr=NULL;

    if(head==NULL)

        printf("List is empty\n");

    else

    {

        ptr=head;

        printf("Enter the node that u want to search:");

        scanf("%d",&key);

        do

        {

            if(ptr->data==key)

            {

                flag=1;

                break;

            }

            else

                ptr=ptr->next;

        } while (ptr!=head);

        if(flag==0)

            printf("%d is not found\n",key);

        else

            printf("%d is present in the list\n",key);

    }

}

//Count

void count()

{

    int cou=0;

    node \*ptr=NULL;

    if(head==NULL)

        printf("List is empty\n");

    else

    {

        ptr=head;

        do

        {

          cou++;

          ptr=ptr->next;

        } while (ptr!=head);

        printf("No.of nodes in the list=%d\n",cou);

    }

}

//main

void main()

{

int ch;

char rep='y';

do

{

 printf(".....Menu.....\n1.Create a list\n2.Display Forward\n");

 printf("3.Insert first\n4. Insert end\n5.Insert After\n");

 printf("6.Delete first\n7.Delete Last\n8.Delete After\n");

 printf("9.Search\n10.Count\n11.Display Backwards\n");

 printf("Select one option:");

 scanf("%d",&ch);

 switch(ch)

 {

     case 1:

        create();break;

     case 2:

        display\_forward(); break;

    case 11:

        display\_rev(); break;

    case 3:

        insert\_first();break;

    case 4:

        insert\_end();break;

    case 5:

        insert\_after();break;

    case 6:

        delete\_first();break;

    case 7:

        delete\_last();break;

    case 8:

        delete\_after();break;

    case 9:

        search();break;

    case 10:

        count();break;

    default:

        printf("Invalid option\n");

 }

 printf("Do u want to repeat y/n:");

 scanf(" %c",&ch);

}while(ch=='y'||ch=='Y');

}