**Program No. 6**

Objective:-WAP for various operation performed on Circular Singly Linked list

#include<stdio.h>

#include<stdlib.h>

#include<conio.h>

struct node\*head=NULL;

struct node

{

int data;

struct node\*link;

};

struct node\*Aakash\_createnewnode()

{

struct node \*newNode;

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

printf("Enter new node=\n");

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

newNode->link=NULL;

return newNode;

}

void printLinkedlist(struct node \*ptr)

{

ptr=head;

while(ptr->link!=head)

{

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

ptr=ptr->link;

}

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

}

void createcircularsinglylinkedlistbyaddingnodeatfront()

{

struct node\*newNode,\*ptr;

if(head==NULL)

{

printf("NO! node in the linked list=\n");

head=Aakash\_createnewnode();

head->link=head;

}

else

{

ptr=head;

while(ptr->link!=head)

{

ptr=ptr->link;

}

newNode=Aakash\_createnewnode();

newNode->link=head;

ptr->link=newNode;

head=newNode;

}

}

void createcircularsinglylinkedlistbyaddingnodeatend()

{

struct node\*newNode,\*ptr;

if(head==NULL)

{

printf("NO! node in the linked list=\n");

head=Aakash\_createnewnode();

head->link=head;

}

else

{

ptr=head;

while(ptr->link!=head)

{

ptr=ptr->link;

}

newNode=Aakash\_createnewnode();

newNode->link=head;

ptr->link=newNode;

}

}

void createcircularsinglylinkedlistbyaddingnodeatanyposition(int key)

{

struct node\*newNode,\*ptr;

if(head==NULL)

{

printf("NO! node in the linked list=\n");

head=Aakash\_createnewnode();

head->link=head;

}

else

{

ptr=head;

while(ptr->data!=key)

{

ptr=ptr->link;

}

newNode=Aakash\_createnewnode();

newNode->link=ptr->link;

ptr->link=newNode;

printLinkedlist(ptr);

}

}

void deletenode()

{

struct node\* ptr,\*ptr1;

if (head==NULL)

{

printf("No! there is no node in the list");

exit(1);

}

else

{

ptr=head;

while(ptr->link!=head)

{

ptr=ptr->link;

}

ptr1=head;

head=head->link;

ptr->link=head;

free(ptr1);

}

if(head->link==head)

{

{

printf("There is only one node & no extra node to delete in the list");

exit(1);

}

}

}

void deletenodeatend()

{

struct node\* ptr,\*myptr;

if (head==NULL)

{

printf("No! there is no node in the list");

}

else

{

ptr=head;

while(ptr->link!=head)

{

myptr=ptr;

ptr=ptr->link;

}

myptr->link=head;

free(ptr);

}

if(head->link==head)

{

{

printf("There is only one node & no extra node to delete in the list");

exit(1);

}

}

}

void deletenodeatanyposition(int key)

{

struct node\* ptr=head,\*myptr;

if (head->link==head)

{

printf("No! there is no node in the list");

exit(1);

}

else

{

while(ptr->data!=key)

{

if(ptr->link==head)

{

printf("No! there is no node found in the list");

exit(1);

}

else

{

myptr=ptr;

ptr=ptr->link;

}

}

if(ptr->link==head)

{

myptr->link=head;

free(ptr);

}

else

{

if(ptr==head)

{

myptr=head;

while(myptr->link!=head)

myptr=myptr->link;

head=head->link;

myptr->link=head;

free(ptr);

}

else

{

if(ptr->link==head)

{

myptr->link=head;

}

else

{

myptr->link=ptr->link;

}

free(ptr);

}

}

}

}

void main()

{

int n,m,o;

char choice;struct node \*ptr;int key;

printf("Enter the value between (1-2)=\n");

printf("\n\nFor Creating Circular Singly Linked List Press 1:");

printf("\n\nFor Deleting Node From Circular Singly Linked List Press 2:\n\n");

scanf("%d",&n);

printf("\n");

switch(n)

{

case 1:

{

printf("Enter the value between (1-3)=\n");

printf("\n\nFor Creating Circular Singly Linked List By Adding Node At Front Press 1:");

printf("\n\nFor Creating Circular Singly Linked List By Adding Node At End Press 2:");

printf("\n\nFor Creating Circular Singly Linked List By Adding Node At Any Position Press 3:");

scanf("%d",&m);

printf("\n");

switch(m)

{

case 1:

{

do

{

createcircularsinglylinkedlistbyaddingnodeatfront();

printf("Do u want to add further node=\n\n");

choice=getch();

}

while(choice=='Y'||choice=='y');

printf("Your Linked List is in order of=\n\n");

printLinkedlist(head);

getch();

break;

}

case 2:

{

do

{

createcircularsinglylinkedlistbyaddingnodeatend();

printf("Do u want to add further node=\n\n");

choice=getch();

}

while(choice=='Y'||choice=='y');

ptr=head;

printf("Your Linked List is in order of=\n\n");

printLinkedlist(ptr);

getch();

break;

}

case 3:

{

do

{

createcircularsinglylinkedlistbyaddingnodeatanyposition(key);

printf("\nEnter the value of key=\n\n");

scanf("%d",&key);

printf("Do u want to add further node=\n\n");

choice=getch();

}

while(choice=='Y'||choice=='y');

printf("Your Linked List is in order of=\n\n");

printLinkedlist(head);

getch();

break;

}

}

}

case 2:

{

printf("Enter the value between (1-3)=\n");

printf("\n\nFor Deleting Any Node From Circular Singly Linked List By Front Press 1:");

printf("\n\nFor Deleting Any Node From Circular Singly Linked List By End Press 2:");

printf("\n\nFor Deleting Any Node From Circular Singly Linked List By Any Position Press 3:\n\n");

scanf("%d",&o);

printf("\n");

switch(o)

{

case 1:

{

do

{

createcircularsinglylinkedlistbyaddingnodeatend();

ptr=head;

printLinkedlist(ptr);

printf("\n\nDo u want to add further node=\n");

choice=getch();

}

while(choice=='Y'||choice=='y');

ptr=head;

printf("\nYour Linked List is in order of=\n");

printLinkedlist(ptr);

printf("\n\nYour Deleted Linked List Is=\n");

do

{

deletenode();

ptr=head;

printf("\n");

printLinkedlist(ptr);

printf("\n\nDo u want to delete further node=\n");

choice=getch();

}

while(choice=='Y'||choice=='y');

ptr=head;

printf("\n\nYour Linked List is in order of=\n");

printLinkedlist(ptr);

break;

}

case 2:

{

do

{

createcircularsinglylinkedlistbyaddingnodeatend();

ptr=head;

printLinkedlist(ptr);

printf("\n\nDo u want to add further node=\n");

choice=getch();

}

while(choice=='Y'||choice=='y');

ptr=head;

printf("\nYour Linked List is in order of=\n");

printLinkedlist(ptr);

printf("\n\nYour Deleted Linked List Is=\n");

do

{

deletenodeatend();

ptr=head;

printf("\n");

printLinkedlist(ptr);

printf("\n\nDo u want to delete further node=\n");

choice=getch();

}

while(choice=='Y'||choice=='y');

ptr=head;

printf("\n\nYour Linked List is in order of=\n");

printLinkedlist(ptr);

break;

}

case 3:

{

do

{

createcircularsinglylinkedlistbyaddingnodeatend();

printf("Do u want to add further node=\n\n");

choice=getch();

}

while(choice=='Y'||choice=='y');

ptr=head;

printf("Your Linked List is in order of=\n\n");

printLinkedlist(head);

printf("\nEnter the key=\n");

scanf("%d",&key);

deletenodeatanyposition(key);

puts("\nLinked List after Deletion of key:\n\n");

printLinkedlist(head);

getch();

break;

}

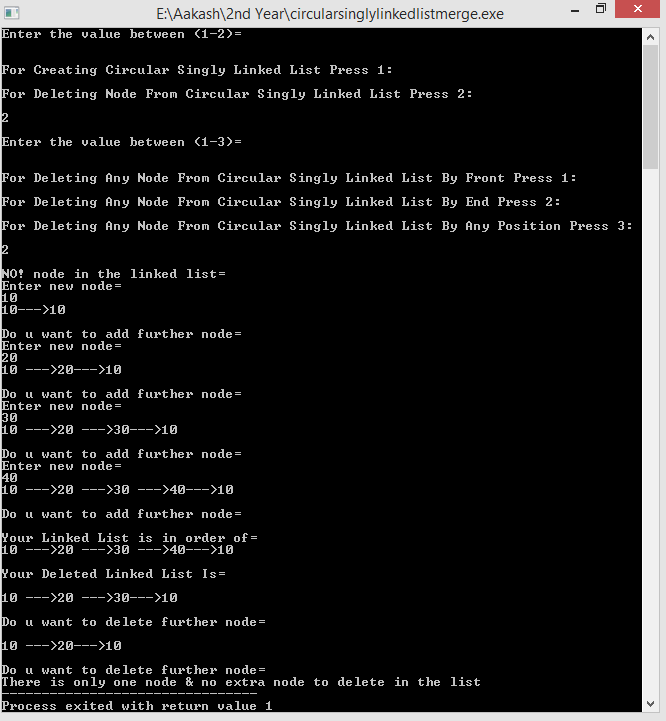
}

}

}

}

**Output:-**



Operation performed on Circular Singly Linked list