# COMPLETE Program on linklist

# Singly Linklist (Traversion)

#include<stdio.h>

#include<stdlib.h>

#include<stdbool.h>

int main()

{

struct node

{

int data;

struct node \*link;

};

struct node \*head,\*newnode,\*temp;

head = 0;

int choice = 1;

int count=0;

##### while(choice)

{

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

printf("Enter data : ");

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

newnode->link = 0;

if(head == 0)

{

head = temp = newnode;

}

else

{

temp->link = newnode;

temp = newnode;

}

printf("Do you want to continue 0/1 ?");

scanf("%d",&choice);

if(choice==1)

{

continue;

}

else

{

break;

}

}

temp = head;

printf("List is : ");

while(temp!=0)

{

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

temp = temp->link;

count++;

}

printf("Null");

printf("\nTotal no. of nodes : %d",count);

return 0;

}

# Singly Linklist (Insertion)

#include<stdio.h>

#include<stdlib.h>

#include<stdbool.h>

int main()

{

struct node

{

int data;

struct node \*link;

};

struct node \*head,\*newnode,\*temp;

head = 0;

int choice = 1;

int count=0;

while(choice)

{

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

printf("Enter data : ");

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

newnode->link = 0;

if(head == 0)

{

head = temp = newnode;

}

else

{

temp->link = newnode;

temp = newnode;

}

printf("Do you want to continue 0/1 ?");

scanf("%d",&choice);

if(choice==1)

{

continue;

}

else

{

break;

}

}

temp = head;

while(temp!=0)

{

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

temp = temp->link;

count++;

}

printf("NULL\n");

("Total no. of nodes : %d",count);

printf

//here display the link list is complete.

// insert at beginning is start from here.

int count1=count+1;

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

printf("\nEnter the data U want to insert at begin : ");

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

newnode->link = head;

head = newnode;

temp = head;

while(temp!=0)

{

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

temp = temp->link;

count++;

}

printf("NULL\n");

printf("Total no. of nodes : %d",count1);

//here display the instert at beginning link list is complete.

// insert at ending is start from here.

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

printf("\nEnter data U want to insert at end : ");

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

newnode->link = 0;

temp =head;

while(temp->link!=0)

{

temp = temp->link;

}

temp->link=newnode;

temp = head;

while(temp!=0)

{

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

temp = temp->link;

count++;

}

printf("NULL\n");

printf("Total no. of nodes : %d",count1+1);

//here display the insert at ending link list is complete.

// insert at after a given location is start from here.

int pos,i=1;

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

printf("\nEnter the position after which U insert at : ");

scanf("%d",&pos);

if(pos>count+1)

{

printf("invalid Position '~' ");

}

else

{

temp=head;

while(i<pos)

{

temp=temp->link;

i++;

}

printf("\nEnter the data U want to insert at : ");

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

newnode->link=temp->link;

temp->link=newnode;

temp=head;

while(temp!=0)

{

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

temp = temp->link;

count++;

}

printf("NULL\n");

printf("Total no. of nodes : %d",count1+2);

}

return 0;

}

# Singly Linklist(Deletion)

#include<stdio.h>

#include<stdlib.h>

struct node

{

int data;

struct node \*link;

};

struct node \*head,\*newnode,\*temp;

int choice = 1;

int count=0;

void create()

{ printf("Create a linklist ^~^ \n");

head = 0;

while(choice)

{

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

printf("Enter data : ");

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

newnode->link = 0;

if(head == 0)

{

head = temp = newnode;

}

else

{

temp->link = newnode;

temp = newnode;

}

printf("Do you want to continue 0/1 ? ");

scanf("%d",&choice);

if(choice==1)

{

continue;

}

else

{

break;

}

}

}

displayNode()

{

temp = head;

printf("Created linklist \n");

while(temp!=0)

{

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

temp = temp->link;

count++;

}

printf("NULL\n");

printf("\nTotal no. of nodes : %d",count);

}

deleteNode()

{ struct node \*prev\_node,\*next\_node;

int choice1,pos,i=1;

printf("\nWant do delete something ^~^\n1.delete at beg : \n2. delete at end : \n3.delete at specified Position : \n4. Not want to delete anything : \n");

scanf("%d",&choice1);

switch(choice1)

{

case 1:

temp=head;

head=head->link;

free(temp);

break;

case 2:

temp=head;

while(temp->link!=0)

{

prev\_node=temp;

temp=temp->link;

}

if(temp==head)

{

head=0;

}

else

{

prev\_node->link=0;

}

free(temp);

break;

case 3:

temp=head;

printf("Enter the Position : ");

scanf("%d",&pos);

while(i<pos-1)

{

temp=temp->link;

i++;

}

next\_node=temp->link;

temp->link=next\_node->link;

free(next\_node);

break;

case 4:

break;

}

}

int main()

{

create();

displayNode();

deleteNode();

displayNode();

return 0;

}

Singly Linklist(Searching)

#include<stdio.h>

#include<stdlib.h>

#include<stdbool.h>

struct node

{

int data;

struct node \*link;

}\*temp,\*head,\*newnode;

CreateLL()

{

head = 0;

int choice = 1;

while(choice)

{

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

printf("Enter data : ");

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

newnode->link = 0;

if(head == 0)

{

head = temp = newnode;

}

else

{

temp->link = newnode;

temp = newnode;

}

printf("Do you want to continue 0/1 ?");

scanf("%d",&choice);

if(choice==1)

{

continue;

}

else

{

break;

}

}

}

Display()

{

int count=0;

temp = head;

printf("\nList is : ");

while(temp!=0)

{

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

temp = temp->link;

count++;

}

printf("NULL");

printf("\nTotal no. of nodes : %d",count);

}

Searching()

{

struct node\*temp;

int num,cho=1,coun=1;

temp=head;

while(cho==1)

{

printf("\nWhich Element You Want to Search : ");

scanf("%d",&num);

printf("element is : %d",num);

while(temp!=0)

{

if(temp->data==num)

{

printf("\nSuccess'~'\nElement is found at Position :%d",coun);

}

temp=temp->link;

coun++;

}

printf("\nDo You want to Search any Element 0/1 : ");

scanf("%d",&cho);

}

}

int main()

{

CreateLL();

Display();

Searching();

return 0;

}

# Circular Linklist(Traversion)

#include<stdio.h>

#include<stdlib.h>

struct node

{

int data;

struct node \*next;

}\*tail,\*temp,\*newnode;

void CreateCLL()

{

int choice = 1;

printf("Create a circular linklist ^~^\n");

while(choice)

{

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

printf("Enter data : ");

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

newnode->next=0;

if(tail==0)

{

tail=newnode;

tail->next=newnode;

}

else

{

newnode->next=tail->next;

tail->next=newnode;

tail=newnode;

}

printf("Do U want to continue 1/0 : ");

scanf("%d",&choice);

}

printf("\nFor Conformation list starts From : (%d",tail->next->data);

printf(")\n");

}

Display()

{

int count=1;

if(tail==0)

{

printf("\nList is Empty \*~\* ");

}

else

{

temp=tail->next;

printf("\nCreated Circular Linklist : ");

while(temp->next!=tail->next)

{

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

temp = temp->next;

count++;

}

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

printf("\nCount Value : %d",count);

}

}

int main()

{

CreateCLL();

Display();

return 0;

}

# Circular Linklist(Insertion)

#include<stdio.h>

#include<stdlib.h>

struct node

{

int data;

struct node \*next;

}\*tail,\*temp,\*newnode;

void CreateCLL()

{

int choice = 1;

printf("Create a circular linklist ^~^\n");

while(choice)

{

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

printf("Enter data : ");

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

newnode->next=0;

if(tail==0)

{

tail=newnode;

tail->next=newnode;

}

else

{

newnode->next=tail->next;

tail->next=newnode;

tail=newnode;

}

printf("Do U want to continue 1/0 : ");

scanf("%d",&choice);

}

printf("\nFor Conformation list starts From : (%d",tail->next->data);

printf(")\n");

}

Display()

{

int count=1;

if(tail==0)

{

printf("\nList is Empty \*~\* ");

}

else

{

temp=tail->next;

printf("\nCreated Circular Linklist : ");

while(temp->next!=tail->next)

{

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

temp = temp->next;

count++;

}

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

printf("\nCount Value : %d",count);

}

}

InsertatBeg()

{

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

printf("Enter data : ");

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

newnode->next=0;

if(tail==0)

{

tail = newnode;

tail->next=newnode;

}

else

{

newnode->next=tail->next;

tail->next=newnode;

}

Display();

}

InsertatEnd()

{

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

printf("Enter data : ");

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

newnode->next=0;

if(tail==0)

{

tail = newnode;

tail->next=newnode;

}

else

{

newnode->next=tail->next;

tail->next=newnode;

tail=newnode;

}

Display();

}

InsertatPos()

{

int pos,i=1,l;

printf("Enter the position : ");

scanf("%d",&pos);

l=20;

if(pos<1||pos>l)

{

printf("Invalid Position \*~\* ");

}

else if(pos==1)

{

InsertatBeg();

}

else

{

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

printf("\nEnter data : ");

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

newnode->next=0;

temp=temp->next;

while(i<pos-1)

{

temp=temp->next;

i++;

}

newnode->next=temp->next;

temp->next=newnode;

}

Display();

}

InsertNode()

{

int choice1=1,choice2 = 1;;

printf("\nWant do Insert something ^~^ ");

while(choice2)

{

printf("\n1. Insert at beg : \n2. Insert at end : \n3. Insert at specified Position : \n4. Not want to Insert anything : \n");

scanf("%d",&choice1);

switch(choice1)

{

case 1:

InsertatBeg();

break;

case 2:

InsertatEnd();

break;

case 3:

InsertatPos();

break;

case 4:

break;

}

printf("\nDo U want to continue 1/0 : ");

scanf("%d",&choice2);

}

}

int main()

{

int l;

printf("Enter the Maximum no. of Nodes for list : ");

scanf("%d",&l);

CreateCLL();

Display();

InsertNode();

Display();

return 0;

}

# Circular Linklist(Deletion)

#include<stdio.h>

#include<stdlib.h>

struct node

{

int data;

struct node \*next;

}\*tail,\*temp,\*newnode;

void CreateCLL()

{

int choice = 1;

printf("Create a circular linklist ^~^\n");

while(choice)

{

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

printf("Enter data : ");

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

newnode->next=0;

if(tail==0)

{

tail=newnode;

tail->next=newnode;

}

else

{

newnode->next=tail->next;

tail->next=newnode;

tail=newnode;

}

printf("Do U want to continue 1/0 : ");

scanf("%d",&choice);

}

printf("\nFor Conformation list starts From : (%d",tail->next->data);

printf(")\n");

}

Display()

{

int count=1;

if(tail==0)

{

printf("\nList is Empty \*~\* ");

}

else

{

temp=tail->next;

printf("\nCreated Circular Linklist : ");

while(temp->next!=tail->next)

{

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

temp = temp->next;

count++;

}

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

printf("\nCount Value : %d",count);

}

}

DeleteFromBeg()

{

temp=temp->next;

if(tail==0)

{

printf("\nList is Empty '~' ");

}

else if(temp->next==temp)

{

tail=0;

free(temp);

}

else

{

tail->next=temp->next;

free(temp);

}

}

DeleteFromEnd()

{

struct node \*current,\*previous;

current=tail->next;

if(tail==0)

{

printf("\nList is Empty '~' ");

}

else if(current->next==current)

{

tail=0;

free(current);

}

else

{

while(current->next!=tail->next)

{

previous=current;

current=current->next;

}

previous->next=tail->next;

tail=previous;

free(current);

}

}

DeleteAtPos()

{

struct node\*current,\*nextnode;

int pos,i=1,l;

printf("Enter the position : ");

scanf("%d",&pos);

l=20;

if(pos<1||pos>l)

{

printf("Invalid Position \*~\* ");

}

else if(pos==1)

{

DeleteFromBeg();

}

else

{

current=tail->next;

while(i<pos-1)

{

current=current->next;

i++;

}

nextnode=current->next;

current->next=nextnode->next;

free(nextnode);

}

Display();

}

DeleteNode()

{

int choice1=1,choice=1;

while(choice)

{

printf("\nWant do Delete something ^~^ ");

printf("\n1. Delete at beg : \n2.Delete at end : \n3. Delete at specified Position : \n");

scanf("%d",&choice1);

switch(choice1)

{

case 1:

DeleteFromBeg();

Display();

break;

case 2:

DeleteFromEnd();

Display();

break;

case 3:

DeleteAtPos();

break;

}

printf("\nDo U want to continue 1/0 : ");

scanf("%d",&choice);

}

}

int main()

{

CreateCLL();

Display();

DeleteNode();

Display();

return 0;

}

# Circular Linklist(Searching)

#include<stdio.h>

#include<stdlib.h>

struct node

{

int data;

struct node \*next;

}\*head;

void CreateCLL()

{

struct node\*temp,\*newnode;

int choice = 1;

printf("Create a circular linklist ^~^\n");

while(choice)

{

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

printf("Enter data : ");

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

newnode->next=0;

if(head==0)

{

head = temp=newnode;

}

else

{

temp->next=newnode;

temp=newnode;

}

temp->next=head;

printf("Do U want to continue 1/0 : ");

scanf("%d",&choice);

}

}

Display()

{

struct node\*temp,\*newnode;

int count=1;

if(head==0)

{

printf("\nList is Empty \*~\* ");

}

else

{

temp=head;

printf("\nCreated Circular Linklist : ");

while(temp->next!=head)

{

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

temp = temp->next;

count++;

}

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

}

}

Searching()

{

struct node\*temp;

int num,cho=1,coun=1;

temp=head;

while(cho==1)

{

printf("\nWhich Element You Want to Search : ");

scanf("%d",&num);

printf("element is : %d",num);

while(temp!=0)

{

if(temp->data==num)

{

printf("\nSuccess'~'\nElement is found at Position :%d",coun);

break;

}

temp=temp->next;

coun++;

}

printf("\nDo You want to Search any Element 0/1 : ");

scanf("%d",&cho);

}

}

int main()

{

CreateCLL();

Display();

Searching();

return 0;

}

# Doubly Linklist(Traversion)

#include<stdio.h>

#include<stdlib.h>

struct node

{

int data;

struct node \*next;

struct node \*prev;

}\*head,\*tail;

void CreateDLL()

{

struct node\*newnode;

int choice= 1;

head=0;

printf("Create a Doubly linklist ^~^\n");

while(choice)

{

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

printf("Enter data : ");

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

newnode->next=0;

if(tail==0)

{

head=tail=newnode;

}

else

{

tail->next=newnode;

newnode->prev=tail;

tail=newnode;

}

printf("Do U want to continue 1/0 : ");

scanf("%d",&choice);

}

}

Display()

{

struct node\*temp;

int count=1;

temp = head;

printf("\nCreated a Doubly linklist : ");

while(temp->next!=0)

{

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

temp = temp->next;

count++;

}

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

printf("\nTotal no. of nodes : %d",count);

}

int main()

{

CreateDLL();

Display();

return 0;

}

# Doubly Linklist(Insertion)

#include<stdio.h>

#include<stdlib.h>

struct node

{

int data;

struct node \*next;

struct node \*prev;

}\*head,\*tail;

void CreateDLL()

{

struct node\*newnode;

int choice= 1;

head=0;

printf("Create a Doubly linklist ^~^\n");

while(choice)

{

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

printf("Enter data : ");

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

newnode->next=0;

if(tail==0)

{

head=tail=newnode;

}

else

{

tail->next=newnode;

newnode->prev=tail;

tail=newnode;

}

printf("Do U want to continue 1/0 : ");

scanf("%d",&choice);

}

}

Display()

{

struct node\*temp;

int count=1;

temp = head;

printf("\nCreated a Doubly linklist : ");

while(temp->next!=0)

{

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

temp = temp->next;

count++;

}

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

printf("\nTotal no. of nodes : %d",count);

}

//count the no. of nodes

int Getcount(struct node \*head)

{

// Base case

if(head==0)

return 0;

//count is 1 + count of remaining list

return 1+Getcount(head->next);

}

InsertAtBeg()

{

struct node\*newnode;

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

printf("Enter data : ");

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

newnode->next=0;newnode->prev=0;

head->prev=newnode;

newnode->next=head;

head=newnode;

}

InsertAtEnd()

{

struct node\*newnode;

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

printf("Enter data : ");

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

newnode->next=0;newnode->prev=0;

tail->next=newnode;

newnode->prev=tail;

tail=newnode;

}

InsertAtPos()

{

struct node\*newnode;

int pos,i=1,l;

printf("Enter the position : ");

scanf("%d",&pos);

l=Getcount(head);

if(pos<1||pos>l)

{

printf("Invalid Position \*~\* ");

}

else if(pos==1)

{

InsertAtBeg();

}

else if (pos==l)

{

InsertAtEnd();

}

else

{

struct node\*temp;

temp=head;

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

printf("\nEnter data : ");

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

while(i<pos-1)

{

temp=temp->next;

i++;

}

newnode->prev=temp;

newnode->next=temp->next;

temp->next=newnode;

newnode->next->prev=newnode;

}

}

InsertNode()

{

int choice1=1,choice2 = 1;;

printf("\nWant do Insert something ^~^ ");

while(choice2)

{

printf("\n1. Insert at beg : \n2. Insert at end : \n3. Insert at specified Position : \n4. Not want to Insert anything : \n");

scanf("%d",&choice1);

switch(choice1)

{

case 1:

InsertAtBeg();

Display();

break;

case 2:

InsertAtEnd();

Display();

break;

case 3:

InsertAtPos();

Display();

break;

case 4:

break;

}

printf("\nDo U want to continue 1/0 : ");

scanf("%d",&choice2);

}

}

int main()

{

CreateDLL();

Display();

InsertNode();

Display();

return 0;

}

# Doubly Linklist(Deletion)

#include<stdio.h>

#include<stdlib.h>

struct node

{

int data;

struct node \*next;

struct node \*prev;

}\*head,\*tail;

void CreateDLL()

{

struct node\*newnode;

int choice= 1;

head=0;

printf("Create a Doubly linklist ^~^\n");

while(choice)

{

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

printf("Enter data : ");

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

newnode->next=0;

if(tail==0)

{

head=tail=newnode;

}

else

{

tail->next=newnode;

newnode->prev=tail;

tail=newnode;

}

printf("Do U want to continue 1/0 : ");

scanf("%d",&choice);

}

}

Display()

{

struct node\*temp;

int count=1;

temp = head;

printf("\nCreated a Doubly linklist : ");

while(temp->next!=0)

{

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

temp = temp->next;

count++;

}

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

printf("\nTotal no. of nodes : %d",count);

}

//count the no. of nodes

int Getcount(struct node \*head)

{

// Base case

if(head==0)

return 0;

//count is 1 + count of remaining list

return 1+Getcount(head->next);

}

DeleteAtBeg()

{

struct node\*temp;

if(head==0)

{

printf("List is Empty \*~\* ");

}

else

{

temp=head;

head=head->next;

head->prev=0;

free(temp);

}

}

DeleteAtEnd()

{

struct node\*temp;

if(tail==0)

{

printf("List is Empty \*~\* ");

}

else

{

temp=tail;

tail->prev->next=0;

tail=tail->prev;

free(temp);

}

}

DeleteAtPos()

{

struct node\*temp;

int pos,i=1,l;

temp=head;

printf("Enter the position : ");

scanf("%d",&pos);

l=Getcount(head);

if(pos<1||pos>l)

{

printf("Invalid Position \*~\* ");

}

else if(pos==1)

{

DeleteAtBeg();

}

else if(pos==l)

{

DeleteAtEnd();

}

else

{

while(i<pos)

{

temp=temp->next;

i++;

}

temp->prev->next=temp->next;

temp->next->prev=temp->prev;

free(temp);

}

}

DeleteNode()

{

int choice1=1,choice2 = 1;;

printf("\nWant do Insert something ^~^ ");

while(choice2)

{

printf("\n1. Delete at beg : \n2. Delete at end : \n3. Delete from specified Position : \n4. Not want to Delete anything : \n");

scanf("%d",&choice1);

switch(choice1)

{

case 1:

DeleteAtBeg();

Display();

break;

case 2:

DeleteAtEnd();

Display();

break;

case 3:

DeleteAtPos();

Display();

break;

case 4:

break;

}

printf("\nDo U want to continue 1/0 : ");

scanf("%d",&choice2);

}

}

int main()

{

CreateDLL();

Display();

DeleteNode();

Display();

return 0;

}

# Doubly Linklist(Searching)

#include<stdio.h>

#include<stdlib.h>

struct node

{

int data;

struct node \*next;

struct node \*prev;

}\*head,\*tail;

void CreateDLL()

{

struct node\*newnode;

int choice= 1;

head=0;

printf("Create a Doubly linklist ^~^\n");

while(choice)

{

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

printf("Enter data : ");

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

newnode->next=0;

if(tail==0)

{

head=tail=newnode;

}

else

{

tail->next=newnode;

newnode->prev=tail;

tail=newnode;

}

printf("Do U want to continue 1/0 : ");

scanf("%d",&choice);

}

}

Display()

{

struct node\*temp;

int count=1;

temp = head;

printf("\nCreated a Doubly linklist : ");

while(temp->next!=0)

{

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

temp = temp->next;

count++;

}

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

printf("\nTotal no. of nodes : %d",count);

}

Searching()

{

struct node\*temp;

int num,cho=1,coun=1;

temp=head;

while(cho==1)

{

printf("\nWhich Element You Want to Search : ");

scanf("%d",&num);

printf("element is : %d",num);

while(temp!=0)

{

if(temp->data==num)

{

printf("\nSuccess'~'\nElement is found at Position :%d",coun);

break;

}

temp=temp->next;

coun++;

}

printf("\nDo You want to Search any Element 0/1 : ");

scanf("%d",&cho);

}

}

int main()

{

CreateDLL();

Display();

Searching();

return 0;

}

# Doubly Circular Linklist(Traversion)

#include<stdio.h>

#include<stdlib.h>

struct node

{

int data;

struct node \*next;

struct node \*prev;

}\*head,\*tail;

void CreateDLL()

{

struct node\*newnode;

int choice= 1;

head=0;

printf("Create a Doubly Circular linklist ^~^\n");

while(choice)

{

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

printf("Enter data : ");

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

newnode->next=0;

if(head==0)

{

head=tail=newnode;

head->next=head;

head->prev=head;

}

else

{

tail->next=newnode;

newnode->prev=tail;

newnode->next=head;

head->prev=newnode;

tail=newnode;

}

printf("Do U want to continue 1/0 : ");

scanf("%d",&choice);

}

printf("\nFor Conformation list : \tStarts From : (%d",tail->next->data);

printf(")\tAnd End At : (%d",tail->data);

printf(")\n");

}

Display()

{

struct node\*temp;

int count=1;

temp = head;

printf("\nCreated a Doubly Circular linklist : ");

while(temp!=tail)

{

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

temp = temp->next;

count++;

}

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

printf("\nTotal no. of nodes : %d",count);

}

//count the no. of nodes

int Getcount(struct node \*head)

{

// Base case

if(head==0)

return 0;

//count is 1 + count of remaining list

return 1+Getcount(head->next);

}

int main()

{

CreateDLL();

Display();

return 0;

}

# Doubly Circular Linklist(Insertion)

#include<stdio.h>

#include<stdlib.h>

struct node

{

int data;

struct node \*next;

struct node \*prev;

}\*head,\*newnode,\*tail,\*temp;

void CreateDCLL()

{

int choice= 1;

head=0;

printf("Create a Doubly linklist ^~^\n");

while(choice)

{

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

printf("Enter data : ");

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

newnode->next=0;

if(head==0)

{

head=tail=newnode;

head->next=head;

head->prev=head;

}

else

{

tail->next=newnode;

newnode->prev=tail;

newnode->next=head;

head->prev=newnode;

tail=newnode;

}

printf("Do U want to continue 1/0 : ");

scanf("%d",&choice);

}

printf("\nFor Conformation list : \tStarts From : (%d",tail->next->data);

printf(")\tAnd End At : (%d",tail->data);

printf(")\n");

}

Display()

{

struct node\*temp;

int count=1;

temp = head;

printf("\nCreated a Doubly Circular linklist : ");

while(temp!=tail)

{

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

temp = temp->next;

count++;

}

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

printf("\nTotal no. of nodes : %d",count);

}

//count the no. of nodes

int Getcount(struct node \*head)

{

// Base case

if(head==0)

return 0;

//count is 1 + count of remaining list

return 1+Getcount(head->next);

}

InsertatBeg()

{

struct node\*temp;

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

printf("Enter data : ");

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

newnode->next=0;

if(head==0)

{

head=tail=newnode;

newnode->prev=tail;

newnode->next=head;

}

else

{

newnode->next=head;

head->prev=newnode;

newnode->prev=tail;

tail->next=newnode;

head=newnode;

}

Display();

}

InsertatEnd()

{

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

printf("Enter data : ");

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

newnode->next=0;

if(head==0)

{

head=tail=newnode;

newnode->prev=tail;

newnode->next=head;

}

else

{

newnode->prev=tail;

tail->next=newnode;

newnode->next=head;

head->prev=newnode;

tail=newnode;

}

Display();

}

InsertatPos()

{

struct node\*temp;

temp=head;

int pos,l,i=1;

printf("Enter the Position : ");

scanf("%d",&pos);

l=Getcount(head);

if(pos<1||pos>l)

{

printf("\nInvalid Position \*~\*");

}

else if(pos==1)

{

InsertatBeg();

}

else if(pos==l)

{

InsertatEnd();

}

else

{

newnode->next=0;newnode->prev=0;

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

printf("Enter data : ");

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

while(i<pos-1)

{

temp=temp->next;

i++;

}

newnode->prev=temp;

newnode->next=temp->next;

temp->next->prev=newnode;

temp->next=newnode;

}

}

InsertNode()

{

int choice1=1,choice2 = 1;;

printf("\nWant do Insert something ^~^ ");

while(choice2)

{

printf("\n1. Insert at beg : \n2. Insert at end : \n3. Insert at specified Position : \n4. Not want to Insert anything : \n");

scanf("%d",&choice1);

switch(choice1)

{

case 1:

InsertatBeg();

break;

case 2:

InsertatEnd();

break;

case 3:

InsertatPos();

break;

case 4:

break;

}

printf("\nDo U want to continue 1/0 : ");

scanf("%d",&choice2);

}

}

int main()

{

CreateDCLL();

Display();

InsertNode();

Display();

return 0;

}

# Doubly Circular Linklist(Deletion)

#include<stdio.h>

#include<stdlib.h>

struct node

{

int data;

struct node \*next;

struct node \*prev;

}\*head,\*newnode,\*tail,\*temp;

void CreateDCLL()

{

int choice= 1;

head=0;

printf("Create a Doubly linklist ^~^\n");

while(choice)

{

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

printf("Enter data : ");

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

newnode->next=0;

if(head==0)

{

head=tail=newnode;

head->next=head;

head->prev=head;

}

else

{

tail->next=newnode;

newnode->prev=tail;

newnode->next=head;

head->prev=newnode;

tail=newnode;

}

printf("Do U want to continue 1/0 : ");

scanf("%d",&choice);

}

printf("\nFor Conformation list : \tStarts From : (%d",tail->next->data);

printf(")\tAnd End At : (%d",tail->data);

printf(")\n");

}

Display()

{

struct node\*temp;

int count=1;

temp = head;

printf("\nCreated a Doubly Circular linklist : ");

while(temp!=tail)

{

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

temp = temp->next;

count++;

}

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

printf("\nTotal no. of nodes : %d",count);

}

//count the no. of nodes

DeleteFromBeg()

{

temp=head;

if(head==0)

{

printf("\nList is Empty '~' ");

}

else if(temp->next==temp)

{

head=tail=0;

free(temp);

}

else

{

head=head->next;

head->prev=tail;

tail->next=head;

free(temp);

}

}

DeleteFromEnd()

{

temp=tail;

if(tail==0)

{

printf("\nList is Empty '~' ");

}

else if(tail->next==tail)

{

head=tail=0;

free(temp);

}

else

{

tail=tail->prev;

tail->next=head;

head->prev=tail;

free(temp);

}

}

int Getcount(struct node \*head)

{

// Base case

if(head==0)

return 0;

//count is 1 + count of remaining list

return 1+Getcount(head->next);

}

DeleteAtPos()

{

temp=head;

int pos,i=0;

printf("Enter the position : ");

scanf("%d",&pos);

int l=Getcount(head);

if(pos<1||pos>l)

{

printf("Invalid Position \*~\* ");

}

else if(pos==1)

{

DeleteFromBeg();

}

else if(pos==l)

{

DeleteFromEnd();

}

else

{

while(i<pos-1)

{

temp=temp->next;

i++;

}

temp->prev->next=temp->next;

temp->next->prev=temp->next->prev;

free(temp);

}

Display();

}

DeleteNode()

{

int choice1=1,choice=1;

while(choice)

{

printf("\nWant do Delete something ^~^ ");

printf("\n1. Delete at beg : \n2.Delete at end : \n3. Delete at specified Position : \n");

scanf("%d",&choice1);

switch(choice1)

{

case 1:

DeleteFromBeg();

Display();

break;

case 2:

DeleteFromEnd();

Display();

break;

case 3:

DeleteAtPos();

break;

}

printf("\nDo U want to continue 1/0 : ");

scanf("%d",&choice);

}

}

int main()

{

CreateDCLL();

Display();

DeleteNode();

Display();

return 0;

}

# Doubly Circular Linklist(Searching)

#include<stdio.h>

#include<stdlib.h>

struct node

{

int data;

struct node \*next;

struct node \*prev;

}\*head,\*tail;

void CreateDCLL()

{

struct node\*newnode;

int choice= 1;

head=0;

printf("Create a Doubly Circular linklist ^~^\n");

while(choice)

{

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

printf("Enter data : ");

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

newnode->next=0;

if(head==0)

{

head=tail=newnode;

head->next=head;

head->prev=head;

}

else

{

tail->next=newnode;

newnode->prev=tail;

newnode->next=head;

head->prev=newnode;

tail=newnode;

}

printf("Do U want to continue 1/0 : ");

scanf("%d",&choice);

}

printf("\nFor Conformation list : \tStarts From : (%d",tail->next->data);

printf(")\tAnd End At : (%d",tail->data);

printf(")\n");

}

Display()

{

struct node\*temp;

int count=1;

temp = head;

printf("\nCreated a Doubly Circular linklist : ");

while(temp!=tail)

{

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

temp = temp->next;

count++;

}

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

printf("\nTotal no. of nodes : %d",count);

}

//count the no. of nodes

int Getcount(struct node \*head)

{

// Base case

if(head==0)

return 0;

//count is 1 + count of remaining list

return 1+Getcount(head->next);

}

Searching()

{

struct node\*temp;

int num,cho=1,coun=1;

temp=head;

while(cho==1)

{

printf("\nWhich Element You Want to Search : ");

scanf("%d",&num);

printf("element is : %d",num);

while(temp!=0)

{

if(temp->data==num)

{

printf("\nSuccess'~'\nElement is found at Position :%d",coun);

break;

}

temp=temp->next;

coun++;

}

printf("\nDo You want to Search any Element 0/1 : ");

scanf("%d",&cho);

}

}

int main()

{

CreateDCLL();

Display();

Searching();

return 0;

}

# *COMPLETE*

Sobhasaria group of instution

Data structure&algrothium

Vishal Chhipa

19esgcs074