**רשימה מקושרת דו כיוונית-**

#include <stdio.h>

#include <stdlib.h>

typedef struct node

{

int data;

struct node \*next;

struct node \*prev;

}Node;

Node \* addToFirst(Node \*head,int data);

Node \* addToLast(Node \*head,int data);

Node \* createNode(int data);

void printList(Node \*head);

void printReverse(Node \*head);

void print(Node \*head);

void main()

{

Node \*head=NULL;

//Node \*temp;

//head=createNode(7);

//temp=createNode(8);

//head->next=temp;

//temp->prev=head;

head=addToFirst(head,2);

head=addToFirst(head,5);

head=addToFirst(head,9);

head=addToFirst(head,3);

print(head);

}

void print(Node \*head)

{

Node \*p=head;

if (head==NULL)

return;

while(p!=NULL)

{

printf("%d\t",p->data);

if (p->next!=NULL)

{

p=p->next->next;

if (p==NULL)

return;

printf("%d\t",p->data);

p=p->prev;

}

else

p=p->next;

}

}

void printReverse(Node \*head)

{

Node \*p=head;

printf("NULL");

if (head==NULL)

return;

for(;p->next!=NULL;p=p->next)

{

}

while(p!=NULL)

{

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

p=p->prev;

}

printf("NULL");

}

void printList(Node \*head)

{

Node \*p=head;

printf("NULL");

if (head==NULL)

return;

while(p!=NULL)

{

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

p=p->next;

}

printf("NULL");

}

Node \* addToFirst(Node \*head,int data)

{

Node \*temp=createNode(data);

if (head==NULL)

return temp;

temp->next=head;

head->prev=temp;

return temp;

}

Node \* addToLast(Node \*head,int data)

{

Node \*temp=createNode(data);

Node \*p=head;

if (head==NULL)

return temp;

while(p->next!=NULL)

p=p->next; // for(p=head;p->next!=NULL;p=p->next);

p->next=temp;

temp->prev=p;

return head;

}

Node \* createNode(int data)

{

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

temp->data=data;

temp->next=NULL;

temp->prev=NULL;

return temp;

}