**פונקציות כלליות-**

#include <stdio.h>

#include <stdlib.h>

typedef struct node

{

int data;

struct node \*next;

}Node;

Node \*createNode(int data);

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

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

void addToLast1(Node \*\*head,int data);

void printList(Node \*head);

void printRec(Node \*head);

Node \* createList();

void createList2(Node \*\*);

Node \*createSortedList();

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

void freeList(Node \*head);

void freeListRec(Node \*head);

void main()

{

Node \*head;

head=createSortedList();

printList(head);

freeList(head);

}

void printList(Node \*head)

{

Node \*p=head;

while(p!=NULL)

{

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

p=p->next;

}

printf("NULL");

}

void printRec(Node \*head)

{

if (head==NULL)

{

printf("NULL");

return;

}

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

printRec(head->next);

}

void freeListRec(Node \*head)

{

if (head==NULL)

return;

freeListRec(head->next);

free(head);

}

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

{

Node \*p=head;

Node \*temp=createNode(data);

if (head==NULL)

return temp;

while(p->next!=NULL)

p=p->next;

p->next=temp;

return head;

}

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

{

Node \*temp=createNode(data);

temp->next=head;

return temp;

}

Node \*createNode(int data)

{

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

temp->data=data;

temp->next=NULL;

return temp;

}

void addToLast1(Node \*\*head,int data)

{

Node \*p=\*head;

Node \*temp=createNode(data);

if (\*head==NULL)

{

\*head=temp;

return;

}

while(p->next!=NULL)

p=p->next;

p->next=temp;

}

Node \* createList()

{

Node \*head=NULL;

Node \* temp;

Node \*p;

int num;

scanf("%d",&num);

while(num!=-1)

{

p=head;

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

temp->data=num;

temp->next=NULL;

if (head==NULL)

head=temp;

else

{

while(p->next!=NULL)

p=p->next;

p->next=temp;

}

scanf("%d",&num);

}

return head;

}

void createList2(Node \*\*head)

{

int num;

\*head=NULL;

scanf("%d",&num);

while(num!=-1)

{

addToLast1(head,num);

scanf("%d",&num);

}

}

Node \*createSortedList()

{

Node \*head=NULL;

int num;

scanf("%d",&num);

while(num!=-1)

{

head=addSorted(head,num);

scanf("%d",&num);

}

return head;

}

void freeList(Node \*head)

{

Node \*p=head;

while(p!=NULL)

{

p=p->next;

free(head);

head=p;

}

}

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

{

Node \*temp=createNode(data);

Node \*first=head,\*second;

if (head==NULL)

return temp;

if (data<head->data)

{

head=addToFirst(head,data);

return head;

}

while(first->next!=NULL)

{

second=first->next;

if (data>=first->data && data<=second->data)

{

first->next=temp;

temp->next=second;

return head;

}

first=first->next;

}

first->next=temp;

return head;

}