C-Program to implement LINKED LIST

( insert-front, delete-front, display, order-list & delete-info)

#include<stdio.h>

#include<conio.h>

#include<alloc.h>

#include<process.h>

struct node

{

int info;

struct node \*link;

};

typedef struct node \*NODE;

NODE getnode()

{

NODE x;

x=(NODE)malloc(sizeof(struct node));

if(x==NULL)

{

printf("mem full\n");

exit(0);

}

return x;

}

void freenode(NODE x)

{

free(x);

}

NODE insert\_front(NODE first,int item)

{

NODE temp;

temp=getnode();

temp->info=item;

temp->link=NULL;

if(first==NULL)

return temp;

temp->link=first;

first=temp;

return first;

}

NODE delete\_front(NODE first)

{

NODE temp;

if(first==NULL)

{

printf("list is empty cannot delete\n");

return first;

}

temp=first;

temp=temp->link;

printf("item deleted at front-end is=%d\n",first->info);

free(first);

return temp;

}

NODE order\_list(int item,NODE first)

{

NODE temp,prev,cur;

temp=getnode();

temp->info=item;

temp->link=NULL;

if(first==NULL) return temp;

if(item<first->info)

{

temp->link=first;

return temp;

}

prev=NULL;

cur=first;

while(cur!=NULL&&item>cur->info)

{

prev=cur;

cur=cur->link;

}

prev->link=temp;

temp->link=cur;

return first;

}

NODE delete\_info(int key,NODE first)

{

NODE prev,cur;

if(first==NULL)

{

printf("list is empty\n");

return NULL;

}

if(key==first->info)

{

cur=first;

first=first->link;

freenode(cur);

return first;

}

prev=NULL;

cur=first;

while(cur!=NULL)

{

if(key==cur->info)break;

prev=cur;

cur=cur->link;

}

if(cur==NULL)

{

printf("search is unsuccessfull\n");

return first;

}

prev->link=cur->link;

printf("key deleted is %d",cur->info);

freenode(cur);

return first;

}

void display(NODE first)

{

NODE temp;

if(first==NULL)

printf("list empty cannot display items\n");

for(temp=first;temp!=NULL;temp=temp->link)

{

printf("%d\n",temp->info);

}

}

void main()

{

int item,choice,key;

NODE first=NULL;

clrscr();

for(;;)

{

printf("\n 1:Insert\_front\t 2:Delete\_front\t ");

printf(" 3:Order\_list\t 4:Delete\_info\t 5:Display\_list\t 6:Exit\n");

printf("enter the choice:\t");

scanf("%d",&choice);

switch(choice)

{

case 1:printf("enter the item at front-end:\t");

scanf("%d",&item);

first=insert\_front(first,item);

break;

case 2:first=delete\_front(first);

break;

case 3:printf("enter the item to be inserted in ordered\_list:\t");

scanf("%d",&item);

first=order\_list(item,first);

break;

case 4:printf("enter the key to be deleted:\t");

scanf("%d",&key);

first=delete\_info(key,first);

break;

case 5:display(first);

break;

default:exit(0);

break;

}

}

}