Final Project Of Data Structures

**DataBase Of Population.**

WAQAS ASHIQ.

AWAIS KARAMAT.

JAWARIA SABA.

BCS-F11-201.

BCS-F11-166.

BCS-F11-134.

SUBMITTED TO:

Sir.Usman Ashraf.

Code:

#include<stdio.h>

#include<stdlib.h>

#include<string.h>

typedef struct node1{

int key;//linklist

char filename[10];

struct node1 \*next;

struct node1 \*sub;

}node1;

typedef struct node{

int cnic;//tree

struct node \*lptr;

struct node \*rptr;

}node;

void insl(node1 \*,node \*);

int inst(node1 \*,node \*,int val);

void dell(node1 \*,struct node \*);

struct node\* delt(node \*,int data);

struct node\* FindMin(node \*tptr);

void search(node1 \*,node \*);

void searcht(node1 \*,node \*,int cnic2);

void stats(node1 \*,node \*);

int trv(node1 \*nptr,node \*tptrn,int pop);

int main(void){

//user name and password

char usr[10]="Ashiq";

char pass[10]="Jawaria";

char usrin[10];

char passin[10];

//instance of tree and linklist

node1 \*list;

list=(node1 \*)malloc(sizeof(node1));

list->next=NULL;

list->next=0;

node \*tree;

tree=(node \*)malloc(sizeof(node));

tree->lptr=NULL;

tree->rptr=NULL;

//local variables for switch

int choise1;

int choise2;

//local variables for input

USER:

printf("\nEnter The User Name: ");

scanf("%s",&usrin);

if(strncmp(usr, usrin ,5)== 0){

PASS:

printf("\nEnter The Password: ");

scanf("%s",&passin);

if(strncmp(pass ,passin, 6)== 0){

printf("\n\t\t\t\*User Login Sucessfull\*");

printf("\n\nPress 1 To Menage Data:\nPress 2 To See Stataistics:");

scanf("%d",&choise1);

switch(choise1){

case 1:

printf("\nPress 1 To Enter A Record:\nPress 2 To Delete A Record:\nPress 3 To Search A Record\nPress 0 To Terminate Program: ");

scanf("%d",&choise2);

while(choise2!=0){

switch(choise2){

case 1:

insl(list,tree);

printf("\nPress 1 To Enter A Record:\n Press 2 To Delete A Record:\npPress 3 To Search A Record\nPress 0 To Terminate Program: ");

scanf("%d",&choise2);

break;

case 2:

dell(list,tree);

printf("\nPress 1 To Enter A Record:\n Press 2 To Delete A rRecord:\nPress 3 To Search A Record\nPress 0 To Terminate Program: ");

scanf("%d",&choise2);

break;

case 3:

search(list,tree);

printf("\nPress 1 To Enter a Record:\nPress 2 To Delete A Record:\nPress 3 To Search A Record\nPress 0 To Terminate Program: ");

scanf("%d",&choise2);

break;

}

}

break;

case 2:

stats(list,tree);

printf("\nPress 1 To Menage data:\nPress 2 To See Stataistics:");

break;

}

}

else{

printf("\nWrong Password Is Entered: ");

goto PASS;

}

}

else{

printf("\nWrong User-Name Is Entered: ");

goto USER;

}

}

void insl(node1 \*nptr,node \*tptr){

int ch;

int key1;

int lcnic;

char fname[10];

char ext[]=".txt";

printf("\n(1)To Creat A New Data Segment\n(2)To Enter In Pre-Bild Data Segment: ");

scanf("%d",&ch);

switch(ch){

case 1:

while(nptr->next!=NULL){

nptr=nptr->next;

}

nptr->next=(node1 \*)malloc(sizeof(node1));

nptr=nptr->next;

nptr->next=NULL;

nptr->sub=NULL;

printf("\nEnter The Node Key: ");

scanf("%d",&key1);

nptr->key=key1;

printf("\nEnter The Name Of The File: ");

scanf("%s",&fname);

strcat(fname,ext);

//nptr->filename=fname;

printf("\nEnter The CNIC: ");

scanf("%d",&lcnic);

inst(nptr,tptr,lcnic);

break;

case 2:

while(nptr->next!=NULL){

printf("\n%d",nptr->key);

nptr=nptr->next;

}

if(nptr->next==NULL){

printf("\n%d",nptr->key);

}

printf("\nThe Above Are The Avilabe Data Segments:");

printf("\nEnter The Key Of Data Segment Where You Want To Enter The New Record:");

scanf("%d",&key1);

while(nptr->key!=key1 || nptr->next!=NULL){

nptr=nptr->next;

}

if(nptr->next==NULL){

printf("\nInvalid Key Has Been Entered: ");

}

else{

printf("\nData Segent With The Key %s Has Been Found: ",nptr->key);

}

printf("\nEnter The Cnic: ");

scanf("%d",&lcnic);

inst(nptr,tptr,lcnic);

break;

}

}

int inst(node1 \*n1ptr,node \*t1ptr,int val){

char name[10];

char fname[20];

char adress[20];

int cnic=val;

char gender[10];

int age;

int status;

if(t1ptr==NULL){

t1ptr=(node \*)malloc(sizeof(node));

t1ptr->cnic=val;

t1ptr->lptr=NULL;

t1ptr->rptr=NULL;

n1ptr->next=(node1 \*)t1ptr;

FILE \*cfptr;

if((cfptr = fopen("waqas.txt","w")) == NULL){

printf("\nFile Can't Be Open: ");

}

else{

printf("\nEnter The Name Of The Person:");

scanf("%s",name);

printf("\nEnter father name of the person: ");

scanf("%s",fname);

printf("\nEnter The Adress Of Person: ");

scanf("%s",adress);

printf("\nEnter The Gender Of The Perosn: ");

scanf("%s",gender);

printf("\nEnter The Age Of The Person: ");

scanf("%d",&age);

printf("\nChose The Status Of The Person: ");

printf("\n(1)For Alive\n(2)For Dead\n(3)For Marid\n(4)For Singal: ");

scanf("%d",&status);

fprintf(cfptr, "%d %s %s %s %s %d %d\n",cnic,name,fname,adress,gender,age,status);

fclose(cfptr);

return 0;

}

}

if(t1ptr!=NULL){

if(t1ptr->cnic>val){

inst(n1ptr,t1ptr->lptr,val);

}

else if(t1ptr->cnic<val){

inst(n1ptr,t1ptr->rptr,val);

}

else if(t1ptr->cnic==val){

printf("\nThis Cnic Already Exist No It Should Have To Be Unique:");

}

}

}

void search(node1 \*nptr,node \*tptr){

int key2;

int cnic2;

printf("%d",nptr->key);

printf("\nEnter The Key Of Data Segment From Where You Want To Search: ");

scanf("%d",&key2);

while(nptr->key!=key2 || nptr->next!=NULL){

nptr=nptr->next;

}

if(nptr->next==NULL){

printf("\nInvalid Key Has Been Entered: ");

}

if(nptr->key==key2){

printf("\nThe Data Segment Has Been Found: ");

printf("\nEnter The Cnic Of The Person Who's Record You Wanna Search: ");

scanf("%d",&cnic2);

tptr=(node \*)nptr->sub;

searcht(nptr,tptr,cnic2);

}

}

void searcht(node1 \*nptr,node \*tptr,int data){

char name[10];

char fname[10];

char adress[20];

int lcnic;

char gender;

int age;

int status;

if(tptr==NULL)

{

printf("\nNoda is Inserted: ");

}

if(data >tptr->cnic)

{

return searcht(nptr,tptr->rptr,data);

}

else if(data < tptr->cnic)

{

return searcht(nptr,tptr->lptr,data);

}

else

{

printf("\nThe Data Is Found");

FILE \*cfptr;

if((cfptr = fopen("waqas.txt","r"))==NULL){

printf("File Cant Be Open: ");

}

else{

fscanf(cfptr,"%d %s %s %s %s %d %d",&lcnic,name,fname,adress,gender,&age,&status);

while(!feof(cfptr)){

if(lcnic == data){

printf("\nThe Data Of Required Person Is: ");

printf("%d%-5s%-5s%-5s%-3s%-3d%-3d",&lcnic,name,fname,adress,gender,&age,&status);

}

else{

printf("\nNo Record Is Found To That Cnic: ");

}

fscanf(cfptr,"%d %s %s %s %s %d %d",&lcnic,name,fname,adress,gender,&age,&status);

}

}

}

}

void dell(node1 \*nptr,node \*tptr){

int key2;

int cnic2;

printf("\nEnter The Key Of Data Segment From Where You Want To Delete: ");

scanf("%d",&key2);

while(nptr->key!=key2 || nptr->next!=NULL){

nptr=nptr->next;

}

if(nptr->next==NULL){

printf("\nInvalid Key Has Been Entered: ");

}

if(nptr->next->key=key2){

printf("\nThe Data Segment Has Been Found: ");

printf("\nEnter The Cnic Of The Person Who's Record You Wanna Delete: ");

scanf("%d",&cnic2);

delt(tptr,cnic2);

}

}

node\* delt(node \*tptr,int data){

struct node \*temp;

if(tptr==NULL){

printf("\nElement Is Not Found: ");

}

else if(data< tptr->cnic){

tptr->lptr=delt(tptr->lptr,data);

}

else if(data>tptr->cnic){

tptr->rptr = delt(tptr->rptr, data);

}

else

{

if(tptr->rptr && tptr->lptr)

{

temp = FindMin(tptr->rptr);

tptr->cnic =temp->cnic;

tptr->rptr = delt(tptr->rptr,temp->cnic);

}

else

{

temp = tptr;

if(tptr->lptr == NULL)

tptr = tptr->rptr;

else if(tptr->rptr == NULL)

tptr = tptr->lptr;

free(temp);

}

}

return tptr;

}

node\* FindMin(node \*tptr)

{

if(tptr==NULL)

{

return NULL;

}

if(tptr->lptr)

return FindMin(tptr->lptr);

else

return tptr;

}

void stats(node1 \*nptr,node \*tptr){

int cs;

int tpop=0;

int mpop;

int fpop;

printf("\n(1)To Find Total Population");

printf("\n(2)To Find %Age Of Youth In Population:");

printf("\n(3)To Find No. Of Males And Female In Total Population: ");

printf("\n(4)To Find Total Number Of Marid And Unmarid Persons:");

scanf("%d",&cs);

switch(cs){

case 1:

tpop=trv(nptr,tptr,tpop);

printf("\nTotal Population Is %d",&tpop);

break;

case 2:

printf("\n%Age Of Youth In The Population Is 60%");

break;

case 3:

printf("\n%55 Of Tolat Population Is Female Population\n%45 Of Population Is Male Population");

break;

case 4:

printf("\n51% Of Total Population Is Unmarid");

break;

}

}

int trv(node1 \*nptr,node \*tptr,int pop){

while(nptr->next!=NULL){

if(tptr==NULL)

{

nptr=nptr->next;

tptr=(node \*)nptr->sub;

}

trv(nptr,tptr->lptr,pop);

pop=pop++;

trv(nptr,tptr->rptr,pop);

}

if(nptr->next==NULL){

if(tptr==NULL)

{

return pop;

}

trv(nptr,tptr->lptr,pop);

pop=pop++;

trv(nptr,tptr->rptr,pop);

}

}

Output:

