#include "genap.h"

void createListsekolah(Listsekolah \*LS){ //create list kosong

(\*LS).firstsekolah=NULL;

}

adrsekolah alokasisekolah(int SK, char namaX[50], int jlh){

adrsekolah s=(adrsekolah)malloc(sizeof(Elmtsekolah));

if(s!=NULL){

s->noSK=SK;

s->nama=namaX[50];

s->jlhSiswa=jlh;

s->firstmurid=NULL;

s->nextsekolah=NULL;

}

return s;

}

void insertFirstsekolah(Listsekolah \*LS, adrsekolah S){ // firstAnak diisi NULL (tambahAnak bisa dilakukan belakangan)

S->nextsekolah=(\*LS).firstsekolah;

(\*LS).firstsekolah=S;

}

void insertLastsekolah(Listsekolah \*LS, adrsekolah S){ // firstAnak diisi NULL (tambahAnak bisa dilakukan belakangan)

adrsekolah last=(\*LS).firstsekolah;

while(last->nextsekolah!=NULL){

last=last->nextsekolah;

}

last->nextsekolah=S;

}

int nbElmtsekolah(Listsekolah LS){ //menghitung banyaknya sekolah di dalam ListSekolah

adrsekolah s=(LS).firstsekolah;

int n=0;

while(n!=NULL){

n++;

s=s->nextsekolah;

}

return n;

}

int maxKuotasekolah(Listsekolah Ls){ //mencari jumlah murid terbanyak dari filed jumlah siswa

adrsekolah s=(Ls).firstsekolah;

int max=s->jlhSiswa;

while(s!=NULL){

if(s->jlhSiswa>max){

max=s->jlhSiswa;

}

s=s->nextsekolah;

}

return max;

}

adrAnak alokasimurid(char namaX[50], boolean genderX){

adrAnak s=(adrAnak)malloc(sizeof(Elmtmurid));

if(s!=NULL){

s->nama=namaX[50];

s->gender=genderX;

s->nextmurid=NULL;

}

return s;

}

void tambahmurid(Listsekolah \*LS, adrmurid m, char namasekolah[50]){

adrsekolah a=(\*LS).firstsekolah;

while (a!=NULL){

if(strcmp((a)->nama,namasekolah[50])==0){

break;

}

a=(a)->nextsekolah;

}

if (a==NULL){

printf("Nama Sekolah Tidak Ditemukan\n");

}

adrAnak b=(b)->firstmurid;

if(b==NULL){

b=c;

(b)->nextmurid=NULL;

(a)->firstmurid=b;

}

while ((b)->nextmurid != NULL){

b=(b)->nextmurid;

}

(b)->nextmurid = c;

(c)->nextmurid = NULL;

}

void cetaksekolahanak(Listsekolah LS){ // mencetak data seluruh sekolah dengan murid-muridnya

adrsekolah s=(LS).firstsekolah;

while(s!=NULL){

printf("No SK : %i\n"

"Nama Sekolah : %s\n"

"Jumlah Siswa : %d\n\n"

,s->SK,s->nama,s->jlhSiswa);

adrAnak b=(LS).firstmurid;

while(b!=NULL){

if(b->nextsekolah==s){

printf("Nama : %s\n",b->nama);

if(b->gender==true)

printf("Jenis Kelamin: laki-laki\n");

else

printf("Jenis Kelamin: perempuan\n");

}

b=b->nextmurid;

}

printf("--------------------------------\n");

s=s->nextsekolah;

}

}

boolean searchsekolahBool(Listsekolah LS, int SK){ // mencari sekolah berdasarkan noSK, bila ketemu maka return true

adrsekolah s=(LS).firstsekolah;

while(s!=NULL && s->noSK!=SK){

s=s->nextsekolah;

}

if(s->noSK==SK){

return true;

else

return false;

}

}

adrsekolah searchsekolah(Listsekolah LS, int SK){

// searchPeg: mencari Sekolah berdasarkan noSK, bila ketemu maka return-lah addressnya, bila tidak ketemu maka return NULL

adrsekolah s=(LS).firstsekolah;

while(s->nextsekolah!=NULL && s->noSK!=SK){

s=s->nextsekolah;

}

if(s->noSK==SK){

return s;

else

return NULL;

}

}

void cetakmurid(Listmurid LS, int SK){ // mencetak murid-murid dari sekolah dengan noSK = sk

adrsekolah s=(LS).firstsekolah;

adrAnak b=(LS).firstmurid;

while(a!=NULL&&a->noSK!=SK){

a=a->nextsekolah;

}

if(a==NULL){

printf("SK Sekolah tidak terdapat dalam data\n");

}else{

while(b!=NULL){

if(b->noSK==a){

printf("Nama Murid : %s\n dari sekolah :%s\n SK sekolah %d \n",b->nama,s->nama,b->SK);

}else{

printf("Tidak ditemukan");

}

b=b->nextmurid;

}

}

}

boolean searchmuridBool(Listsekolah LS, char namamurid[50]){// mencari murid berdasarkan nama, bila ketemu maka return true

adrAnak b=(LS).firstmurid;

while(b!=NULL && b->nama=namamurid[50]){

b=b->nextmurid;

}

if(s->nama==namamurid[50]){

return true;

else

return false;

}

}

int searchmurid(Listsekolah LS, char namamurid[50]){

// searchmurid: mencari murid berdasarkan nama, yang pertama ketemu langsung direturn noSK sekolahnya, bila tidak ketemu maka return -1

adrAnak s=(LS).firstmurid;

while(s->nextmurid!=NULL && s->nama!=namamurid[50]){

s=s->nextmurid;

}

if(s->nama==namamurid[50]){

return SK;

else

return -1;

}

}

int nbmurid(Listsekolah LS){ //menghitung banyaknya murid dari semua sekolah di dalam Listsekolah

adrAnak b=(LS).firstmurid;

int n=0;

while(b!=NULL){

n++;

b=b->nextmurid;

}

return n;

}

int nbMan(Listsekolah LS){ //menghitung banyaknya murid laki-laki dari semua sekolah di dalam ListSekolah

adrAnak b=(LS).firstmurid;

int n=0;

while(b!=NULL){

if(b->gender==true)

n++;

b=b->nextAnak;

}

return n;

}

int nbDaughter(Listsekolah LS){ //menghitung banyaknya murid perempuan dari semua sekolah di dalam

int total=nbmurid(LA);

int man=nbMan(LA);

return (total-man);

}

**"Saya mengerjakan tugas ini dengan jujur dan tidak melakukan pelanggaran aturan dari IFLAB.**

**Jika saya terbukti melanggar, maka saya bersedia menerima sanksi yang berlaku."**