*Colocviu PC*

#include <stdio.h>

#include <conio.h>

#include <string.h>

struct catalog{

char nume[60];

int nota;

}catalog[200];

void inscriere(int n){

printf("Dati numele studentului\n");

scanf("%s", &catalog[n].nume);

printf("Dati nota lui disciplina");

scanf("%d", &catalog[n].nota);

}

void inscriere\_nota(int n){

int ok = 0;

int i;

char nume[50];

printf("Dati numele studentului caruia vreti sa ii adaugati nota:");

scanf("%s", &nume);

for(i = 0; i < n; i++)

if(strcmp(nume, catalog[i].nume) == 0) {

printf("Dati nota studentului:");

scanf("%d", &catalog[i].nota);

ok = 1;

break;

}

if (ok == 0)

printf("Studentul nu a fost gasit in baza de date!");

}

void citire\_nota(int n){

int i;

int ok = 0;

char nume[50];

printf("Dati numele studentului caruia vreti sa ii cititi nota:");

scanf("%s", &nume);

for(i = 0; i < n; i++)

if(strcmp(nume, catalog[i].nume) == 0) {

printf("Nota studentului este: %d ", catalog[i].nota);

ok = 1;

break;

}

if (ok == 0)

printf("Studentul nu a fost gasit in baza de date!");

}

void studenti\_alocati(int n){

int i;

int k,l;

char tmp[50];

if(n != 0){

printf("Studentii alocati disciplinei sunt: ");

for(k = 0; k < n; k++) // ordonare alfabetica catalog

for(l = k+1 ; l < n; l++)

if(strcmp(catalog[k].nume, catalog[l].nume) > 0){

strcpy(tmp,catalog[k].nume);

strcpy(catalog[k].nume,catalog[l].nume);

strcpy(catalog[l].nume,tmp);

}

for(i = 0; i < n; i++){

printf("%s, ", catalog[i].nume);

}

printf("\n Numarul studentilor alocati este: %d", n);

}

else printf("Nu sunt studenti inscrisi la materie!");

}

void prezenti(int n){

int contor = 0;

int i;

for(i = 0; i < n; i++)

if(catalog[i].nota > -1){

printf("Studentul %s a fost prezent!", catalog[i].nume);

contor++;

}

printf("\n Numarul studentilor prezenti este: %d", contor);

}

void studenti\_promovati(int n){

int contor = 0;

int i;

for(i = 0; i < n; i++)

if(catalog[i].nota >= 5){

printf("Studentul %s a promovat!", catalog[i].nume);

contor++;

}

printf("\n Numarul studentilor promovati este: %d", contor);

}

void studenti\_nepromovati(int n){

int contor = 0;

int i;

for(i = 0; i < n; i++)

if(catalog[i].nota < 5){

printf("Studentul %s nu a promovat!", catalog[i].nume);

contor++;

}

printf("\n Numarul studentilor nepromovati este: %d", contor);

}

void studenti\_cu\_o\_anumita\_nota(int n){

int contor = 0;

int i;

int nota;

printf("Dati nota pe care doriti sa o cautati in catalog: ");

scanf("%d", &nota);

for(i = 0; i < n; i++)

if(catalog[i].nota == nota){

printf("\nStudentul %s are nota %d!", catalog[i].nume, nota);

contor++;

}

printf("\n Numarul studentilor cu nota %d este: %d", nota, contor);

}

void studenti\_peste\_o\_anumita\_nota(int n){

int contor = 0;

int i;

int nota;

printf("Dati nota: ");

scanf("%d", &nota);

for(i = 0; i < n; i++)

if(catalog[i].nota > nota){

printf("\nStudentul %s are peste nota %d!", catalog[i].nume, nota);

contor++;

}

printf("\n Numarul studentilor peste nota %d este: %d", nota, contor);

}

void studenti\_sub\_o\_anumita\_nota(int n){

int contor = 0;

int i;

int nota;

printf("Dati nota: ");

scanf("%d", &nota);

for(i = 0; i < n; i++)

if(catalog[i].nota < nota){

printf("\nStudentul %s are sub nota %d!", catalog[i].nume, nota);

contor++;

}

printf("\n Numarul studentilor sub nota %d este: %d", nota, contor);

}

int main() {

char raspuns;

char caz;

int n = 0, flag = 1;

do {

printf("1.Adaugati studentul si nota sa in catalog\n2.Inscrieti nota pentru un anumit student\n3.Cititi nota pentru un anumit student\n4.Studenti alocati disciplinei\n5.Studenti prezenti\n6.Studenti promovati\n7.Studenti nepromovati\n8.Studenti cu o anumita nota\n9.Studenti peste o anumita nota\n10.Studenti sub o anumita nota\n");

caz = getch();

switch (caz) {

case '1':

inscriere(n);

n++;

printf("Doriti o noua informatie? Y/N");

raspuns = getch();

if(strchr(raspuns, 'y') )

break;

else return 0;

case '2':

inscriere\_nota(n);

printf("Doriti o noua informatie? Y/N");

raspuns = getch();

if(strchr(raspuns, 'y'))

break;

else return 0;

case '3':

citire\_nota(n);

printf("Doriti o noua informatie? y/N");

raspuns = getch();

if(strchr(raspuns, 'y'))

break;

else return 0;

case '4':

studenti\_alocati(n);

printf("Doriti o noua informatie? y/N");

raspuns = getch();

if(strchr(raspuns, 'y'))

break;

else return 0;

case '5':

prezenti(n);

printf("Doriti o noua informatie? y/N");

raspuns = getch();

if(strchr(raspuns, 'y'))

break;

else return 0;

case '6':

studenti\_promovati(n);

printf("Doriti o noua informatie? y/N");

raspuns = getch();

if(strchr(raspuns, 'y'))

break;

else return 0;

case '7':

studenti\_nepromovati(n);

printf("Doriti o noua informatie? y/N");

raspuns = getch();

if(strchr(raspuns, 'y'))

break;

else return 0;

case '8':

studenti\_cu\_o\_anumita\_nota(n);

printf("Doriti o noua informatie? y/N");

raspuns = getch();

if(strchr(raspuns, 'y'))

break;

else return 0;

case '9':

studenti\_peste\_o\_anumita\_nota(n);

printf("Doriti o noua informatie? y/N");

raspuns = getch();

if(strchr(raspuns, 'y'))

break;

else return 0;

case '10':

studenti\_sub\_o\_anumita\_nota(n);

printf("Doriti o noua informatie? y/N");

raspuns = getch();

if(strchr(raspuns, 'y'))

break;

else return 0;

default:

break;

}

}while(flag);

return 0;

}