//LINGUAGEM C: LISTA COMPLEMENTAR 6

//1

#include <stdio.h>

#include <stdlib.h>

#include <string.h>

struct Aluno {

char nome[50];

float nota;

char logradouro[50];

long int numero;

};

void ordenarCrescente(struct Aluno \*a, int n);

void ordenarDecrescente(struct Aluno \*a, int n);

struct Aluno maiorNota(struct Aluno \*a, int n);

struct Aluno menorNota(struct Aluno \*a, int n);

void ordemAlfabetica(struct Aluno \*a,int n);

int main() {

int i,n;

char op;

struct Aluno \*a;

struct Aluno maior,menor;

printf("\n Entre com a quantidade de alunos: ");

scanf("%d",&n);

a=(struct Aluno \*)malloc(n\*sizeof(struct Aluno));

if(a==NULL) {

printf("\nERRO - MEMORIA INSUFICIENTE !!!");

exit(1);

}

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

getchar();

printf("\nNome do aluno %d: ",i+1);

gets(a[i].nome);

printf("\nNota do aluno %d: ",i+1);

scanf("%f",&a[i].nota);

}

while(1) {

printf("\n Escolha uma das opÃ§Ãµes abaixo: ");

printf("\na â€“ Imprimir os dados dos alunos por ordem crescente de notas ");

printf("\nb â€“ Imprimir os dados dos alunos por ordem decrescente de notas ");

printf("\nc â€“ Imprimir os dados dos alunos com a maior nota ");

printf("\nd â€“ Imprimir os dados dos alunos com a menor nota ");

printf("\ne â€“ Imprimir os dados dos alunos que moram no mesmo logradouro ");

printf("\nf â€“ Imprimir os dados dos alunos por ordem alfabÃ©tica de nomes ");

printf("\ng â€“ Sair ");

printf("\n OpÃ§Ã£o: ");

getchar();

op=getchar();

system("clear");

if(op=='g') {

break;

}

switch(op) {

case 'a': ordenarCrescente(a,n); break;

case 'b': ordenarDecrescente(a,n); break;

case 'c': maior = maiorNota(a,n);

printf("\nDados do aluno com a maior nota: ");

printf("\nNome do aluno: %s",maior.nome);

printf("\nNota do aluno: %.1f",maior.nota); break;

case 'd': menor = menorNota(a,n);

printf("\nDados do aluno com a menor nota: ");

printf("\nNome do aluno: %s",menor.nome);

printf("\nNota do aluno: %.1f",menor.nota); break;

case 'e': printf("\n Entre com o logradoruro: ");

case 'f': ordemAlfabetica(a,n); break;

default: printf("\nOpcao Invalida!\n"); break;

}

}

free(a);

printf("\n\n");

return 0;

}

//OrdenaÃ§Ã£o Crescente

void ordenaCrescente (struct Aluno \*a, int n) {

int i,j;

struct Aluno aux;

for(i=0;i<n-1;i++)

for(j=i+1;j<n;j++)

if(a[i].nota > a[j].nota) {

aux = a[i];

a[i] = a[j];

a[j] = aux;

}

printf("\nDados dos alunos ordenados crescentemente pela nota: ");

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

printf("\nNome do aluno %d: %s",i+1,a[i].nome);

printf("\nNota do aluno %d: %.1f",i+1,a[i].nota);

}

}

//OrdenaÃ§Ã£o Decrescente

void ordenaDecrescente (struct Aluno \*a, int n) {

int i,j;

struct Aluno aux;

for(i=0;i<n-1;i++)

for(j=i+1;j<n;j++)

if(a[i].nota < a[j].nota) {

aux = a[i];

a[i] = a[j];

a[j] = aux;

}

printf("\nDados dos alunos ordenados decrescentemente pela nota: ");

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

printf("\nNome do aluno %d: %s",i+1,a[i].nome);

printf("\nNota do aluno %d: %.1f",i+1,a[i].nota);

}

}

//Maior Nota

struct Aluno maiorNota(struct Aluno \*a, int n) {

int i;

float aux;

aux = a[0].nota;

for(i=1;i<n;i++)

if(a[i].nota > aux)

aux = a[i].nota;

for(i=1;i<n;i++)

if(a[i].nota==aux) {

printf("\nNome do aluno: %s",maior[i].nome);

printf("\nNota do aluno: %.1f",maior[i].nota);

}

}

//Menor Nota

struct Aluno menorNota(struct Aluno \*a, int n) {

int i;

float aux;

aux = a[0].nota;

for(i=1;i<n;i++)

if(a[i].nota > aux)

aux = a[i].nota;

for(i=1;i<n;i++)

if(a[i].nota==aux) {

printf("\nNome do aluno: %s",maior[i].nome);

printf("\nNota do aluno: %.1f",maior[i].nota);

}

}

//Ordem Alfabetica de Nomes

void ordemAlfabetica(struct Aluno \*a,int n) {

int i,j;

struct Aluno aux;

for(i=0;i<n-1;i++)

for(j=i+1;j<n;j++)

if(strcmp(a[i].nome,a[j].nome) > 0) {

aux=a[i];

a[i]=a[j];

a[j]=aux;

}

}