#include <stdio.h>

#include <string.h>

#include <stdlib.h>

typedef struct Alunos{

char matricula[30],nome[100];

int faltas,flag;

float media;

}talunos;

void cadastrar(FILE \* arq\_turma, int pos);

void exibir(FILE \* arq\_turma,int pos);

void ordenar\_Cres(FILE \*arq\_turma,int pos);

void preecherVet(FILE \*arq\_turma,struct Alunos vet[],int pos);

void ordenar\_Decre(FILE \*arq\_turma,int pos);

void ordenar\_Faltas(FILE \*arq\_turma,int pos);

void consultar(FILE \*arq\_turma,int pos);

int buscar(FILE \* arq\_turma,int posFim, char matricula[]);

FILE \* abrirArquivo(char nome[]){

FILE \* arq;

arq = fopen(nome, "r + b");

if (arq == NULL)

arq = fopen(nome, "w + b");

return arq;

}

int main(){

FILE \*arq\_turma;

talunos turma;

int pos,resp;

char op;

arq\_turma = abrirArquivo("Turma.dat");

if(arq\_turma == NULL)

printf("Erro de Criacao ou Abertura do arquivo!\n");

else{

do{

system("cls");

printf("Menu de opcoes:\n1-Cadastrar novo aluno\n2-Exibir dados dos alunos da turma\n");

printf("3-Ordenar crescente - por matricula\n4-Ordenar Decrescente - por medias\n");

printf("5-Ordenar Crescente - por quantidade de faltas\n6-Consultar Aluno\n7-Sair");

printf("\nDigite a opcao:\n");

scanf("%c",&op);fflush(stdin);fflush(stdin);

fseek(arq\_turma,0,2);

pos = ftell(arq\_turma);

pos = pos / sizeof(talunos);

switch(op){

case '1': printf("Cadastrando:\n");

if(pos > 60)

printf("A turma ja Atingiu a sua cota maxima!\n");

else

cadastrar(arq\_turma, pos);

system("pause");

break;

case '2': printf("Dados dos alunos:\n");

exibir(arq\_turma, pos);

system("pause");

break;

case '3': printf("Ordenando Crescentemente - Por matriculas\n");

ordenar\_Cres(arq\_turma,pos);

system("pause");

break;

case '4': printf("Ordenando Decresentemente - Por Medias\n");

ordenar\_Decre(arq\_turma,pos);

system("pause");

break;

case '5': printf("Ordenando Crescentemente - Por faltas\n");

ordenar\_Faltas(arq\_turma,pos);

system("pause");

break;

case '6': printf("Consultando Aluno:\n");

consultar(arq\_turma,pos);

system("pause");

break;

case '7':

break;

}

}while(op != '7');

}

resp = fclose(arq\_turma);

return 0;

}

int buscar(FILE \* arq\_voo, int posFim, char matricula[]){

talunos x,aux,vet[60];

int i=-1, resp,procurado,inicio=0,fim,meio,achou;

fseek(arq\_voo, 0, 0);

while (1){

resp = fread(&x, sizeof(talunos), 1, arq\_voo);

if(x.flag != 1){

if(i == posFim + 1 )

break;

else if(resp != 1){

if (!feof(arq\_voo))

return -2;

else

return -1;

}

else{

i++;

if (strcmp(x.matricula,matricula) == 0)

return i;

}

}

else{

preecherVet(arq\_voo,vet,posFim);

fim = posFim;

do{

meio = (inicio+fim)/2;

if(strcmp(vet[meio].matricula,matricula) == 0){

achou = 1;

break;

}

else if(strcmp(vet[meio].matricula,matricula)>0)

fim = meio - 1;

else{

inicio = meio + 1;

}

}while(inicio <= fim);

if(achou == 1)

return meio;

else

return -1;

}

}

return -1;

}

void cadastrar(FILE \* arq\_turma, int pos){

talunos turma;

int posi,status;

printf("digite a matricula: ");

gets(turma.matricula);fflush(stdin);

posi = buscar(arq\_turma,pos,turma.matricula);

if(posi != -1){

if(posi == -2)

printf("Erro durante a busca da matricula!\n");

else

printf("Matricula existente!\n");

}

else{

printf("Digite o nome: ");

gets(turma.nome);fflush(stdin);

printf("Digite a media: ");

scanf("%f",&turma.media);

printf("Digite a quantidade de faltas: ");

scanf("%i",&turma.faltas);

turma.flag = 0;

fseek(arq\_turma,0,2);

status = fwrite(&turma,sizeof(talunos),1,arq\_turma);

if(status != 1)

printf("Erro durando a gravacao do registro!\n");

else{

printf("Aluno cadastrado com sucesso!\n");

}

}

}

void exibir(FILE \* arq\_turma,int posfim){

talunos vet;

int i=-1,resp;

fseek(arq\_turma,0,0);

while(1){

if(i == posfim + 1)

break;

resp = fread(&vet,sizeof(talunos),1,arq\_turma);

if(resp != 1){

if(!feof)

printf("Erro de leitura durante a exibicao!\n");

break;

}

else{

i++;

printf("Aluno : %s",vet.nome);

printf("\nMatricula : %s",vet.matricula);

printf("\nMedia : %f",vet.media);

printf("\nQuantidade de faltas : %i\n",vet.faltas);

}

}

}

void ordenar\_Cres(FILE \* arq\_turma,int posFim){

int i=0,pos,fim,troca=1,resp;

talunos vet[60],vetArq,aux;

preecherVet(arq\_turma,vet,posFim);

fim = posFim;

while(troca == 1){

troca = 0;

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

if(strcmp(vet[i].matricula,vet[i+1].matricula) > 0){

aux = vet[i];

vet[i] = vet[i+1];

vet[i+1] = aux;

pos = i;

troca = 1;

}

vet[i].flag = 1;

}

fim = fim - 1;

}

fseek(arq\_turma,0,0);

i = 0;

while(i < posFim){

resp = fread(&vetArq,sizeof(talunos),1,arq\_turma);

if(resp != 1)

break;

else{

vetArq = vet[i];

fseek(arq\_turma,i\*sizeof(talunos),0);

resp = fwrite(&vetArq,sizeof(talunos),1,arq\_turma);

i++;

if(resp != 1){

printf("Erro durante o preechimento do arquivo na ordenacao p/ matricula!\n");

break;

}

}

}

if(resp == 1)

printf("Ordenacao por matricula com sucesso !\n");

}

void preecherVet(FILE \*arq\_turma,struct Alunos vet[],int pos){

int i=0,resp;

talunos vetAux;

fseek(arq\_turma,0,0);

while(1){

if(i == pos + 1)

break;

resp = fread(&vetAux, sizeof(talunos),1,arq\_turma);

if(resp != 1){

if(!feof)

printf("Erro durante o preechimento, erro de leitura!");

break;

}

else{

vet[i].faltas=vetAux.faltas;

vet[i].media=vetAux.media ;

strcpy(vet[i].matricula,vetAux.matricula);fflush(stdin);

strcpy(vet[i].nome,vetAux.nome);fflush(stdin);

vet[i]=vetAux;

i++;

}

}

}

void ordenar\_Decre(FILE \*arq\_turma,int pos){

talunos vet[60],vetArq,aux;

int i,j,menor,resp;

preecherVet(arq\_turma,vet,pos);

for(i=0;i<pos-1;i++){

menor = i;

for(j=i+1 ;j <= pos - 1; j++){

if(vet[j].media < vet[menor].media)

menor = j;

}

aux = vet[i];

vet[i] = vet[menor];

vet[menor] = aux;

vet[i].flag = 0;

}

fseek(arq\_turma,0,0);

i = 0;

while(i < pos){

resp = fread(&vetArq,sizeof(talunos),1,arq\_turma);

if(resp != 1)

break;

else{

vetArq = vet[i];

fseek(arq\_turma,i\*sizeof(talunos),0);

resp = fwrite(&vetArq,sizeof(talunos),1,arq\_turma);

i++;

if(resp != 1){

printf("Erro durante o preechimento do arquivo na ordenacao p/ matricula!\n");

break;

}

}

}

if(resp == 1)

printf("Ordenacao por Media com sucesso !\n");

}

void ordenar\_Faltas(FILE \*arq\_turma,int posFim){

int i,j,resp;

talunos vet[59],vetArq,aux;

preecherVet(arq\_turma,vet,posFim);

for(i=1;i<posFim;i++){

aux = vet[i];

j = i - 1;

while(j>=0 && vet[j].faltas > aux.faltas){

vet[j+1] = vet[j];

j = j - 1;

}

vet[j+1] = aux;

}

fseek(arq\_turma,0,0);

i = 0;

while(i < posFim){

resp = fread(&vetArq,sizeof(talunos),1,arq\_turma);

if(resp != 1)

break;

else{

vetArq = vet[i];

fseek(arq\_turma,i\*sizeof(talunos),0);

resp = fwrite(&vetArq,sizeof(talunos),1,arq\_turma);

i++;

if(resp != 1){

printf("Erro durante o preechimento do arquivo na ordenacao p/ matricula!\n");

break;

}

}

}

if(resp == 1)

printf("Ordenacao por Falta com sucesso !\n");

}

void consultar(FILE \*arq\_turma,int posFim){

talunos arq;

int pos, resp;

char matricula[50];

printf("Digite a matricula do aluno : \n");

gets(matricula);

pos = buscar(arq\_turma,posFim, matricula);

if(pos == -1 )

printf("Matricula nao encontrada ! \n");

else if(pos == -2)

printf("Erro de leitura durante o processo de busca ! \n");

else{

fseek(arq\_turma, pos \* sizeof(talunos),0);

resp = fread(&arq,sizeof(talunos),1,arq\_turma);

if(resp != 1)

printf("Erro na leitura do arquivo, durante a exibição!\n");

else{

printf("\nAluno : %s",arq.nome);

printf("\nMedia : %.2f\n",arq.media);

printf("Quantidade de faltas : %i\n",arq.faltas);

}

}

}