Uma instituição oferece cursos em 3 turnos: M🡪 Manhã, T🡪 Tarde, N🡪 Noite e os alunos inscritos em disciplinas EaD, comuns aos cursos, devem fazer a prova presencial, realizada em julho, no turno de seu curso. O comitê de organização da prova e alocação das salas de aulas, precisa saber quantos alunos realizarão as provas em cada um dos turnos.

O arquivo **disciplinas.txt** armazena a quantidade de alunos nas ( no máximo 15) diferentes disciplinas EaD, do seguinte modo:

* Disciplina (inteiro);
* Qt de alunos inscritos na disciplina em cada um dos 3 turnos (3 inteiros)

**Exemplo do arquivo disciplinas.txt:**

|  |  |  |
| --- | --- | --- |
| **1001**  **400 200 1000** |  | **Significado** 🡺 a disciplina 1001 tem 400 alunos inscritos no turno manhã, 200 alunos inscritos no turno tarde e 1000 alunos inscritos no turno noite |

No início do mês de junho, os alunos inscritos em alguma disciplina EaD podem solicitar a troca do turno para a realização da prova. O arquivo texto **solicitacoes.txt** armazena a disciplina, a matrícula do aluno, o turno origem e o turno destino:

* Disciplina (inteiro);
* Matrícula do aluno ( inteiro) turno de origem ( M ou T ou N) turno destino( M ou T ou N)

**Exemplo do arquivo solicitacoes.txt**:

|  |  |  |
| --- | --- | --- |
| **1001 102030 M T** |  | **Significado** 🡺O aluno de matrícula 102030, matriculado no turno Manhã, deseja realizar a prova de 1001 no turno Tarde. |
|  |

Faça um programa MODULARIZADO que, utilizando as informações armazenadas nos 2 arquivos, gere o arquivo **ATUAL.TXT** com a quantidade de alunos que devem realizar as provas em cada um dos turnos em cada um das disciplinas.

**Obrigatoriamente**, o seu programa deve utilizar uma **matriz** para armazenar e manipular a quantidade de alunos em cada turno de cada disciplina e as seguintes funções feitas por você:

1. função **busca()**: implementa o algoritmo de busca sequencial;
2. função **le\_arq\_disciplinas()**: realiza a leitura dos dados do arquivo **disciplinas.txt**, preenchendo os parâmetros recebidos;
3. função **atualiza\_situacao**: uma de suas responsabilidades é fazer a leitura dos dados do arquivo **solicitacoes.txt**, efetivando as operações realizadas;
4. função **gera\_arq\_atual()**: cria o arquivo **ATUAL.TXT** com as informações atualizadas;
5. função **descobre\_pos\_turnos()**: a função recebe o caracter que representa o turno origem e o caracter que representa o turno destino e dois endereços de memória, onde a função irá armazenar o número correspondente ao turno origem e o correspondente ao turno destino, isto é, 0 se for manhã, 1, se for tarde e 2 se for noite.

|  |  |  |
| --- | --- | --- |
| **disciplinas.txt** | **solicitacoes.txt** | **Atual.txt** |
| 1001  100 200 300  1002  200 100 300  2001  1 2 3  2002  3 4 5 | 1001 102030 M T  2001 102030 M T  1001 203010 T M  2002 203010 N M  1002 102323 N T  1001 103443 M N  1002 102124 T N  1001 102124 T N | 1001 99 199 302  1002 200 100 300  2001 0 3 3  2002 4 4 4 |

Gabarito baseado na solução de Camila Kelly Wong Mendonça

#include<stdio.h>

#include<stdlib.h>

#define MAX 15

#define COL 3

int busca(int \*v,int elemento,int qtd)

{

int i;

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

{

if(\*(v+i)==elemento)

return i;

}

return -1;

}

int le\_arq\_disciplinas(int \*v,int m[MAX][COL], int max)

{

FILE \*dis;

int cont=0,i;

dis=fopen("P://Exercicio 1//disciplinas.txt","rt");

if(dis==NULL)

{

printf("Erro abertura arquivo disciplinas.txt \n");

exit(1);

}

while(cont < max && fscanf(dis,"%d",(v+cont))==1)

{

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

{

fscanf(dis,"%d",&m[cont][i]);

}

cont=cont+1;

}

fclose(dis);

return cont;

}

int Acha\_col( char turno)

{

if(turno=='M')

return 0;

else

{

if(turno=='T')

return 1;

else

return 2;

}

}

void descobre\_pos\_turnos(char antes,int \*posantes,char depois,int \*posdepois)

{

\*posantes=acha\_col(antes);

\*posdepois= acha\_col(depois);

}

void atualiza\_situacao(int \*v,int m[MAX][COL],int qtd)

{

int cod,posicao,mat,posantes,posdepois;

char antes,depois;

FILE \*sol;

sol=fopen("P://Exercicio 1//solicitacoes.txt","rt");

if(sol==NULL)

{

printf("Erro abertura arquivo solicitacoes.txt \n");

exit(1);

}

while(fscanf(sol,"%d",&cod)==1)

{

posicao=busca(v,cod,qtd);

fscanf(sol,"%d %c %c",&mat,&antes,&depois);

if (posicao !=-1)

{

descobre\_pos\_turnos(antes,&posantes,depois,&posdepois);

m[posicao][posantes]=m[posicao][posantes]-1;

m[posicao][posdepois]=m[posicao][posdepois]+1;

}

}

fclose(sol);

}

void gera\_arq\_atual(int \*v, int m[][COL],int qtd)

{

FILE \*atual;

int i;

atual=fopen("atual.txt","wt");

if(atual==NULL)

{

printf("Erro abertura arquivo atual.txt \n");

exit(1);

}

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

{

fprintf(atual,"\n%4d",\*(v+i));

for(j=0;i<COL;j++)

fprintf(atual,"%6d",m[i][j]);

}

fclose(atual);

}

int main(void)

{

int v[MAX],m[MAX][COL],qtd;

qtd=le\_arq\_disciplinas(v,m);

atualiza\_situacao(v,m,qtd);

gera\_arq\_atual(v,m,qtd);

return 0;

}