

NOME: .....

- 1) Suponha a existência de um arquivo texto “REQUISICAO.txt” contendo as seguintes informações: Nro da requisição, Data da requisição, nome do paciente, código do exame médico solicitado e CRM do médico. Você deve ler o arquivo texto e carregá-lo numa tabela na memória principal, conforme ilustração abaixo:

**TABREQUISITA**

Nro. Requisição	Data da Requisição	Nome Paciente	Código do Exame solicitado	CRM do médico
100	03/10/2018	Maria Paulina	4352	120
200	03/10/2018	Maria Paulina	5510	120
300	04/10/2018	Josefina	8222	500
400	04/10/2018	Josefina	5670	500
500	04/10/2018	Josefina	5510	500
600	05/10/2018	Pedro	4566	380

- 2) Suponha a existência dos seguintes arquivos de dados desordenados (salvo em memória secundária) e dos seus índices correspondentes (vetor de índice ordenado):

**INDEXAME** (vetor de índice do EXAME MÉDICO – ordenado)

CODIGO inteiro	POSENDER inteiro
----------------	------------------

**EXAMES** (arquivo binário Desordenado – salvo em memória secundária)

CODEXAME Código Exame médico inteiro	DESCRICAO – descrição char[40]
PRECO float	

**Arquivo de dados binário EXAMES (desordenado)**

CODEXAME	DESCRICAO	PRECO
5670	Lipidograma completo	\$ 180,00
4352	Creatinina	\$ 110,00
8222	Urina (microscópico, macroscópico e bioquímico)	\$150,00
5510	uréia	\$ 120,00
4566	Dosagem de ácido úrico	\$ 50,00

**INDMEDICO** (vetor de índice do CRM do MÉDICO – ordenado)

CODIGO inteiro	POSENDER inteiro
----------------	------------------

**MÉDICOS** (arquivo binário Desordenado – salvo em memória secundária)

CRM do médico inteiro	Nome do médico char[40]
Especialidade char [20]	

**Arquivo de dados binário MÉDICOS (desordenado)**

CRM	Nome do médico	Especialidade
380	Dr.Amil Khalil	Cardiologista
500	Dr.Renato Vasconcelos	Clínico geral
120	Dra.Giulia Ferreira	ginecologista

- a) Declare todas as estruturas de dados envolvidas (struct e class). Declare todas as funções necessárias para que o exercício possa ser resolvido (no class) com os seus devidos parâmetros;

NOME: .....

b) Desenvolva uma função para ler o arquivo texto (REQUISICAO.txt). Carregar uma tabela na memória principal contendo todas as requisições de exames dos pacientes (TABREQUISITA);

c) Desenvolva uma função

**int BUSCABINARIAINDEXADA(int codigo, struct vetindice[ ], int tl)**

Você deve procurar um código num vetor de índice que está ordenado. E precisa devolver/retornar o POSENDER. Caso não encontre o código no vetor de índice, deve retornar com -1 (um negativo).

d) Desenvolva uma função RELATREQUISITA. Este relatório deve conter todas as requisições de exames solicitados pelos médicos aos seus pacientes. Para cada requisição solicitada, você deve:

d.1) Buscar o código do exame (CODEXAME) no vetor de índice (Busca Binária Indexada) e, em seguida, fazer a leitura do mesmo no arquivo de EXAMES. Isto vai permitir informar a descrição do exame e o preço do mesmo. Você deve informar no relatório o valor que deve ser pago pelo exame. Totalizar os valores de todos os exames para cada paciente (ver exemplo abaixo).

d.2) Buscar o CRM do médico (CRM) no vetor de índice (Busca Binária Indexada) e, em seguida, fazer a leitura do mesmo no arquivo de MÉDICOS. Informar o nome do médico e a especialidade no relatório.

Paciente: MARIA PAULINA – Data da requisição 03/10/2018

Exames: Creatinina – Preço: \$ 110,00 Dra.Giulia Ferreira – Ginecologista

Uréia - Preço: \$ 120,00 Dra.Giulia Ferreira – Ginecologista

TOTAL..... \$ 230,00

---

Paciente: JOSEFINA – Data da requisição 04/10/2018

Exames: Urina (micro, macro e Bio) – Preço: \$ 150,00 Dr. Renato Vasconcelos – Clínico Geral

Lipidograma completo - Preço: \$ 180,00 Dr. Renato Vasconcelos – Clínico Geral

Uréia.....preço \$ 120,00 Dr. Renato Vasconcelos – Clínico Geral

TOTAL..... \$ 450,00

---

Projeto solicitado 20/abril/2021 – Programação II – Profa.Liliane Jacon –  
Entrega: 26/abril/2021 até 18 horas  
Grupo de 2 alunos. Enviar no email liliane285@gmail.com

NOME: .....

```
#include<stdio.h>
#include<string.h>
#define TF 40
```

```
struct requisicao {
    int nroreq;
    char nome[10];
    int codexame;
    int crm;
};
```

```
struct indice {
    int codigo;
    int posender;
};
```

```
struct exame{
    int codexame;
    char descricao[15];
    float preco;
};
```

```
struct medico{
    int crm;
    char nome[10];
    char especialidade[10];
};
```

```
class Prova2 {
public:
    int tlr, tle, tlm;
    struct indice vetIndMedico[TF];
    struct indice vetIndExame[TF];
    struct requisicao TabReq[TF];
    Prova2();
    int BuscaBinariaIndexada(int codigo, struct indice vetor[],int tl);
    void carregaTabelaRequisicao();
    void relatorio();
    void criaArqExames();
    void criaArqMedicos();
    void criaIndices();
    void exibeMedicos();
    void exibeExames();
    void ordena(struct indice vetIndice[],int tl);
};
```

```
Prova2::Prova2(){
    tlr=tle=tlm=0;
    int i;
    for (i=0;i<TF;i++){
        strcpy(TabReq[i].nome,"");
    }
}
```

Projeto solicitado 20/abril/2021 – Programação II – Profa.Liliane Jacon –  
Entrega: 26/abril/2021 até 18 horas  
Grupo de 2 alunos. Enviar no email liliane285@gmail.com

NOME: .....  
}

```
void Prova2::carregaTabelaRequisicao(){
    FILE *arq = fopen("requisicao.txt","r");
    tlr=0;
    int nu,ce,cc;
    char no[10];
    if (arq==NULL)
        printf("\n erro. não achou o arquivo texto requisicao.txt");
    else {
        while (!feof(arq)){
            fscanf(arq,"%d %d %d %s",&nu,&ce,&cc,&TabReq[tlr].nome);
            TabReq[tlr].nrreq = nu;
            TabReq[tlr].codexame = ce;
            TabReq[tlr].crm= cc;
            tlr++;
        }
        fclose(arq);

        printf("\n tlr %d",tlr);
        int i;
        for (i=0;i<tlr;i++){
            printf("\n i=%d req=%d nome=%s codexame=%d crm=%d ",i,
                TabReq[i].nrreq,TabReq[i].nome,TabReq[i].codexame,TabReq[i].crm);
        }
        printf("\n");
    }
}
```

```
int Prova2::BuscaBinariaIndexada(int codigo, struct indice vetor[],int tl){
    int inicio,fim,meio;
    inicio=0;
    fim=tl-1;
    meio=(inicio+fim)/2;
    while ((inicio<fim)&&(vetor[meio].codigo !=codigo)){
        if (codigo < vetor[meio].codigo)
            fim=meio;
        else inicio = meio+1;
        meio=(inicio+fim)/2;
    }
    if (vetor[meio].codigo ==codigo)
        return vetor[meio].posender;
    else return -1;
}
```

```
void Prova2::criaArqExames(){
    FILE *arqnovo = fopen("exames.bin","w");
    struct exame e;
    printf("\n codigo de exame = 0 finaliza");
    scanf("%d",&e.codexame);
    while(e.codexame!=0){
        printf("\n descricao do exame");
```

Projeto solicitado 20/abril/2021 – Programação II – Profa.Liliane Jacon –  
Entrega: 26/abril/2021 até 18 horas  
Grupo de 2 alunos. Enviar no email liliane285@gmail.com

NOME: .....

```
scanf("%s",&e.descricao);
printf("\n preco exame");
scanf("%f",&e.preco);
fwrite(&e, sizeof(exame),1,arqnovo);
printf("\n codigo de exame = 0 finaliza");
scanf("%d",&e.codexame);
strset(e.descricao, ' ');
}
fclose(arqnovo);
FILE *arq = fopen("exames.bin","r");
printf("\n Exibe os EXAMES CADASTRADOS NO ARQUIVO");
fread(&e,sizeof(exame),1,arq);
while (!feof(arq)){
    printf("\n                Codexame=%d                descricao=%s
preco=%5.2f",e.codexame,e.descricao,e.preco);
    fread(&e,sizeof(exame),1,arq);
}
fclose(arq);
}
void Prova2::exibeExames(){
    struct exame e;
    FILE *arq = fopen("exames.bin","r");
    printf("\n Exibe os EXAMES CADASTRADOS NO ARQUIVO");
    fread(&e,sizeof(exame),1,arq);
    while (!feof(arq)){
        printf("\n                Codexame=%d                descricao=%s
preco=%5.2f",e.codexame,e.descricao,e.preco);
        fread(&e,sizeof(exame),1,arq);
    }
    fclose(arq);
}
void Prova2::criaArqMedicos(){
    FILE *arqnovo = fopen("medicos.bin","w");
    struct medico m;
    printf("\n crm do medico = 0 finaliza");
    scanf("%d",&m.crm);
    while(m.crm!=0){
        do{ printf("\n nome:");
            gets(m.nome);
        }while (strlen(m.nome)==0);
        do{ printf("\n especialidade:");
            gets(m.especialidade);
        }while (strlen(m.especialidade)==0);
        fwrite(&m, sizeof(medico),1,arqnovo);
        printf("\n crm do medico = 0 finaliza");
        scanf("%d",&m.crm);
        strcpy(m.nome,"");
        strcpy(m.especialidade,"");
    }
    fclose(arqnovo);
    FILE *arq = fopen("medicos.bin","r");
    printf("\n Exibe os MEDICOS CADASTRADOS NO ARQUIVO");
    fread(&m,sizeof(medico),1,arq);
```

Projeto solicitado 20/abril/2021 – Programação II – Profa.Liliane Jacon –  
Entrega: 26/abril/2021 até 18 horas  
Grupo de 2 alunos. Enviar no email liliane285@gmail.com

```
NOME: .....
while (!feof(arq)){
    printf("\n CRM=%d nome=%s Especialidade=%s",m.crm,m.nome,m.especialidade);
    fread(&m,sizeof(medico),1,arq);
}
fclose(arq);
}

void Prova2::exibeMedicos(){
    struct medico m;
    FILE *arq = fopen("medicos.bin","r");
    printf("\n Exibe os MEDICOS CADASTRADOS NO ARQUIVO");
    fread(&m,sizeof(medico),1,arq);
    while (!feof(arq)){
        printf("\n CRM=%d nome=%s Especialidade=%s",m.crm,m.nome,m.especialidade);
        fread(&m,sizeof(medico),1,arq);
    }
    fclose(arq);
}

void Prova2::ordena(struct indice vetIndice[],int tl){
    //ordena o vetor de Indice
    indice aux;
    int fim,i;
    for (fim=tl-1;fim>0;fim--){
        for (i=0;i<fim;i++){
            if (vetIndice[i].codigo > vetIndice[i+1].codigo) { //troca
                aux = vetIndice[i];
                vetIndice[i]=vetIndice[i+1];
                vetIndice[i+1]=aux;
            } //if
        } //for J
        printf("\n vetor de Indice ordenado");
        for (i=0;i<tl;i++){
            printf("\n codigo %d posender %d",vetIndice[i].codigo,vetIndice[i].posender);
        }
    }

void Prova2::criaIndices(){
    FILE *arqe=fopen("exames.bin","r");
    FILE *arqm=fopen("medicos.bin","r");
    struct exame e;
    struct medico m;
    tle=0;
    fread(&e,sizeof(exame),1,arqe);
    while(!feof(arqe)){
        vetIndExame[tle].codigo=e.codexame;
        vetIndExame[tle].posender=tle;
        tle++;
        fread(&e,sizeof(exame),1,arqe);
    }
    fclose(arqe);
    int i;
    for (i=0;i<tle;i++){
        printf("\n Codexame %d posender %d",vetIndExame[i].codigo,vetIndExame[i].posender);
        printf("\n ORDENA O INDICE DO EXAME");
        ordena(vetIndExame,tle);
    }
}
```

Projeto solicitado 20/abril/2021 – Programação II – Profa.Liliane Jacon –  
Entrega: 26/abril/2021 até 18 horas  
Grupo de 2 alunos. Enviar no email liliane285@gmail.com

NOME: .....

```
    tlm=0;
    fread(&m,sizeof(medico),1,arqm);
    while(!feof(arqm)){
        vetIndMedico[tlm].codigo=m.crm;
        vetIndMedico[tlm].posender=tlm;
        tlm++;
        fread(&m,sizeof(medico),1,arqm);
    }
    fclose(arqm);
    for (i=0;i<tlm;i++)
        printf("\n Crm %d posender %d",vetIndMedico[i].codigo,vetIndMedico[i].posender);
    printf("\n ORDENA O INDICE DO MEDICO");
    ordena(vetIndMedico,tlm);
}

void Prova2::relatorio(){
    FILE *arqe=fopen("exames.bin","r");
    FILE *arqm=fopen("medicos.bin","r");
    struct medico m;
    struct exame e;
    struct requisicao req;
    printf("\n Relatorio solicitado na Prova");
    int i;
    char nome[10],descricao[10], nomemed[10],especial[10];
    float total=0, preco;
    int cod,crm,posicaoex,posicaomed;

    strcpy(nome,TabReq[0].nome);
    printf("\n Paciente: %s Requisicao:%d ",TabReq[0].nome,TabReq[0].nrreq);

    for (i=0;i<tlr;i++){
        if (strcmp(nome,TabReq[i].nome) !=0 ){// mudou o nome
            printf("\n Total..... $%5.2f\n",total);
            total = 0;
            strcpy(nome,TabReq[i].nome);
            printf("\n Paciente: %s Requisicao:%d ",TabReq[i].nome,TabReq[i].nrreq);
        }
        cod = TabReq[i].codexame;
        posicaoex = BuscaBinariaIndexada(cod,vetIndExame,tle);
        if (posicaoex==-1) printf("\n ERROOO");
        else {
            fseek(arqe,posicaoex*sizeof(exame),SEEK_SET);
            fread(&e, sizeof(exame), 1, arqe);
            strcpy(descricao,e.descricao);
            preco = e.preco;
            total += preco;
        }
        crm = TabReq[i].crm;
        posicaomed = BuscaBinariaIndexada(crm,vetIndMedico,tlm);
        if (posicaomed==-1) printf("\n ERROOO");
        else {
            fseek(arqm,posicaomed*sizeof(medico),SEEK_SET);
            fread(&m, sizeof(medico), 1, arqm);
            strcpy(nomemed,m.nome);
```

Projeto solicitado 20/abril/2021 – Programação II – Profa.Liliane Jacon –  
Entrega: 26/abril/2021 até 18 horas  
Grupo de 2 alunos. Enviar no email liliane285@gmail.com

NOME: .....

```
                strcpy(especial,m.especialidade);
            }
            printf("\n Exame:%s  Preco:%5.2f  Medico:%s %s",descricao,preco,nomemed,especial);

        }
        printf("\n Total..... %5.2f\n",total);
        fclose(arqe);
        fclose(arqm);
    }
}
```

```
main(){
    Prova2 p = Prova2();
    int opcao=0;
    while (opcao != 8){
        printf("\n 1 - Leitura texto e carrega Tabela requisicao");
        printf("\n 2 - Cria os indices");
        printf("\n 3 - Cria arquivo binario de exames");
        printf("\n 4 - Exibe exames cadastrados no arquivo");
        printf("\n 5 - Cria arquivo binario de medicos");
        printf("\n 6 - Exibe medicos cadastrados no arquivo");
        printf("\n 7 - Relatorio");
        printf("\n 8 - sair");
        scanf("%d",&opcao);
        switch(opcao){
            case 1: p.carregaTabelaRequisicao();
                    break;
            case 2: p.criaIndices();
                    break;
            case 3: p.criaArqExames();
                    break;
            case 4: p.exibeExames() ;
                    break;
            case 5: p.criaArqMedicos();
                    break;
            case 6: p.exibeMedicos();
                    break;
            case 7: p.relatorio();
                    break;
        }
    }
}
```