



Q1	
Q2	
Q3	
Nota	

Aluno: _____

Matrícula: _____ Turma: _____

Questão 1) (3.0 pontos) Uma empresa permite que seus funcionários façam chamadas internacionais a partir de seus ramais, mas mantém um registro dessas ligações no arquivo **CI.TXT**. Cada linha do arquivo contém a matrícula de um funcionário (inteiro), o número de chamadas realizadas pelo mesmo (inteiro) e a duração em minutos de cada uma das ligações (real).

Escreva um programa em C que leia o arquivo **CI.TXT** e gere o arquivo **TOTAIS.TXT**. O arquivo **TOTAIS.TXT** deverá conter, para cada funcionário que efetuou chamadas internacionais, uma linha com seu número de matrícula e a duração total de suas chamadas. A última linha do arquivo deve conter a duração total das chamadas internacionais.

Exemplo (apenas um exemplo!):

<u>ci.txt</u>				
1124	2	3.4	2.2	
3095	0			
4127	1	9.5		
9440	3	7.4	15.2	30.8

<u>totais.txt</u>	
1124	5.6
4127	9.5
9440	53.4
	68.5



Aluno: _____

```
#include <stdio.h>

int main(void) {

    FILE *ent, *sai;

    int matr, quant, i;
    float dur, somadur, somatudo;

    ent = fopen("ci.txt","r"); //0.1
    if (ent==NULL) { //0.2
        printf("Nao foi possivel abrir ci.txt\n");
        return 1;
    }

    sai = fopen("totais.txt","w"); //0.1
    if (sai==NULL) { //0.2
        printf("Nao foi possivel abrir totais.txt\n");
        fclose(ent); //0.2
        return 1;
    }

    somatudo = 0.0; //0.1

    while (fscanf(ent,"%d %d",&matr,&quant)==2) { //0.4
        if (quant>0) { //0.2
            somadur = 0.0; //0.1
            fprintf(sai,"%d",matr); //0.1
            for (i=0; i<quant; i++) { //0.4
                fscanf(ent,"%f",&dur); //0.1
                somadur += dur; //0.2
            }
            fprintf(sai," %.1f\n",somadur); //0.1
            somatudo += somadur; //0.2
        }
    }

    fprintf(sai,"%.1f\n",somatudo); //0.1

    fclose(sai); //0.1
    fclose(ent); //0.1

    return 0;
}

//Se aluno optar por usar vetores, -0.5 por limitar o numero de fun-
cionarios possiveis
```



Aluno: _____

Questão 2) (3.0 pontos) Considere que o cadastro dos livros de uma loja virtual compreenda, entre outras, as seguintes informações para cada livro:

Código: um número inteiro;

Assunto: um número inteiro que codifica o assunto do livro

Estoque: quantidade de unidades do livro em estoque

- a) **(1.0 ponto)** Escreva uma função que receba como parâmetros o número de livros cadastrados (n), um vetor de inteiros já preenchido com os códigos dos livros (`livros[]`), e um inteiro representando o código de um livro desejado (`cod`), e que retorne a posição deste livro no vetor. Se o livro não for encontrado, a função deve retornar -1. A função deve ter o seguinte protótipo:

```
int consulta_livro(int n, int livros[], int cod);
```

- b) **(2.0 pontos)** Escreva uma função que receba como parâmetros o número de livros cadastrados (n), os vetores já preenchidos com os códigos dos livros (`livros[]`), com os códigos dos assuntos (`ass[]`), com as quantidades em estoque (`est[]`), e um inteiro representando o código de um livro desejado. Esta função deve mostrar na tela a quantidade em estoque do livro desejado. Caso não haja o livro em estoque, devem ser mostrados os códigos de todos os livros do mesmo assunto. Caso o livro desejado não seja encontrado, a função deve exibir a mensagem "Livro não cadastrado". Esta função deve fazer uso, **OBRIGATORIAMENTE**, da função do item a) e tem o seguinte protótipo:

```
void mostra_estoque(int n, int livros[], int ass[],  
int est[], int cod);
```



Aluno: _____

```
#include <stdio.h>

int consulta_livro(int n, int livros[], int cod) {
    int pos;
    for (pos=0;pos<n;pos++) { //0.2
        if (livros[pos]==cod) //0.4, e se usar else/return: -0.4
            return pos; //0.2
    }
    return -1; //0.2
}

void mostra_estoque(int n,int livros[],int ass[],int est[],int cod) {
    int i,pos;
    pos=consulta_livro(n,livros,cod); //0.2
    if (pos==-1) printf("\nLivro não cadastrado"); //0.5
    else if (est[pos]>0) printf("\nEm estoque: %d",est[pos]); //0.5
    else for (i=0;i<n;i++) //0.3
        if (ass[i]==ass[pos]) //0.3
            printf("\nLivro - %d",livros[i]); //0.2
}
```



Aluno: _____

Questão 3) (4.0 pontos) Uma operadora de telefonia celular oferece diversos planos aos clientes (máximo 20). Os planos diferem em quantidade de minutos livres, valor do minuto excedente e valor da assinatura. O arquivo **PLANOS.TXT** armazena as condições de todos os planos, ou seja, em cada linha deste arquivo há o código de um plano, o valor de sua assinatura, a quantidade de minutos livres neste plano e o valor do minuto excedente.

- a) **(1.0 ponto)** Faça uma função que preencha os vetores código dos planos (**cod[]**), valor das assinaturas (**assin[]**), quantidades de minutos livres (**qtmin[]**) e valores do minuto excedente (**minexc[]**) com os dados armazenados no arquivo **PLANOS.TXT**. A função deve retornar o número de planos efetivamente lidos e ter o seguinte protótipo:

```
int carrega_vetores(int cod[], float assin[],  
                    int qtmin[], float minexc[]);
```

- b) **(2.0 pontos)** Faça uma função que imprima na tela todas as informações sobre cada plano e seu respectivo valor a pagar, dada uma quantidade de minutos desejada (**mindesejados**). A função deve ter o seguinte protótipo:

```
void mostra_planos(int numplanos, int cod[], float assin[],  
                  int qtmin[], float minexc[], int mindesejados);
```

- c) **(1.0 ponto)** Faça um programa que, fazendo uso **OBRIGATÓRIO** das funções dos itens **a)** e **b)**, mostre as informações dos planos e quanto o cliente pagaria em cada um deles, dada uma quantidade de minutos desejada. Inicialmente, seu programa deve carregar em vetores as informações dos planos a partir do arquivo. A seguir, cada cliente digita a quantidade de minutos desejada e o programa mostra todas as informações e o valor a pagar para cada um dos planos. A entrada de dados termina quando for digitada uma quantidade de minutos inválida (<0).

Exemplo (apenas um exemplo!), supondo 4 planos distintos:

```
Minutos Desejados: 100  
10 20.00 100 0.30 Total: R$ 20.00  
20 0.00 0 1.50 Total: R$ 150.00  
30 40.00 300 0.50 Total: R$ 40.00  
40 5.00 0 0.20 Total: R$ 25.00
```

```
Minutos Desejados: 200  
10 20.00 100 0.30 Total: R$ 50.00  
20 0.00 0 1.50 Total: R$ 300.00  
30 40.00 300 0.50 Total: R$ 40.00  
40 5.00 0 0.20 Total: R$ 45.00
```

```
Minutos Desejados: 150  
10 20.00 100 0.30 Total: R$ 35.00  
20 0.00 0 1.50 Total: R$ 225.00  
30 40.00 300 0.50 Total: R$ 40.00  
40 5.00 0 0.20 Total: R$ 35.00
```

```
Minutos Desejados: 900  
10 20.00 100 0.30 Total: R$ 260.00  
20 0.00 0 1.50 Total: R$1350.00  
30 40.00 300 0.50 Total: R$ 340.00  
40 5.00 0 0.20 Total: R$ 185.00
```



Aluno: _____

```
#include <stdio.h>
#include <stdlib.h>
#define MAX 20

int carrega_vetores(int cod[],float assin[],int qtmin[],float minexc[]) {
    FILE *ent;
    int i=0;
    ent = fopen("planos.txt","r"); //0.3, e se nao verificar, -0.2
    if (ent==NULL) {
        printf( "Impossivel abrir planos.txt\n");
        exit(1);
    }
    //0.5, e se nao limitar a MAX(20), -0.2
    while (i<MAX && fscanf(ent,"%d %f %d %f",&cod[i],&assin[i],
        &qtmin[i],&minexc[i]) == 4)
        i++;
    fclose(ent); //0.1
    return i; //0.1
}

void mostra_planos(int numplanos, int cod[], float assin[] ,int qtmin[],
    float minexc[], int mindesejados) {
    float totplano;
    int i, qtminpg;
    for (i=0;i<numplanos;i++) { //0.3
        totplano = assin[i]; //1.4, pelo calculo correto de totplano
        qtminpg = mindesejados - qtmin[i];
        if (qtminpg > 0) totplano += qtminpg*minexc[i];
        printf("%d %.2f %d %.2f - R$ %.2f\n",cod[i],assin[i],qtmin[i],
            minexc[i],totplano); //0.3
    }
}

int main (void) {
    int cod[MAX],qtmin[MAX],numplanos;
    float assin[MAX],minexc[MAX];
    int mindesejados;
    //Se abrir arquivo na main, -0.4 (alem do desconto na letra a)
    numplanos = carrega_vetores(cod,assin,qtmin,minexc); //0.2
    printf("Digite minutos desejados: "); //0.2 - printf / scanf
    scanf("%d",&mindesejados);
    while (mindesejados >=0) { //0.2
        mostra_planos(numplanos,cod,assin,qtmin,minexc,mindesejados); //0.2
        printf("Digite minutos desejados: "); //0.2 - printf / scanf
        scanf("%d",&mindesejados);
    }
    return 0;
}
```