

IF61C—Fundamentos de Programação 1

Lista de exercícios 15 - Registros

1. (*) Crie a estrutura **Tempo**, composto pelos campos **horas**, **minutos** e **segundos**. Então, leia do teclado um tempo expresso em segundos divida-o em horas, minutos e segundos. Armazene o resultado em uma estrutura.
2. (*) Adapte o programa anterior de tal forma a criar uma **função** que recebe como parâmetro de entrada um tempo em segundos e retorna uma estrutura **Tempo**, contendo esse tempo dividido em horas, minutos e segundos.
3. (**) Dados os seguintes campos de um registro: nome, dia de aniversário e mês de aniversário, desenvolver um programa que mostre para cada um dos meses do ano quem são as pessoas que fazem aniversário. Considere um conjunto de 40 pessoas.
4. (**) Uma pessoa cadastrou um conjunto de 15 registros contendo o cnpj da loja, telefone e preço de um eletrodoméstico (crie um laço para preencher estes registros - organizados em uma estrutura vetor - usando a função **rand()**). Desenvolver um programa que permita exibir qual foi a média dos preços cadastrados e uma relação contendo o cnpj e o telefone das lojas cujo preço estava abaixo da média.
5. (***) Um sistema de administração de pessoal mantém as informações referentes aos funcionários de uma empresa de duas formas: em um vetor de ponteiros para dados estruturados do tipo **Funcionario** ou em um vetor contendo dados estruturados do tipo **Funcionario**. Este tipo é descrito a seguir:

```
struct funcionario {  
    int mat; /* matricula do funcionario */  
    char nome[51]; /* nome do funcionario */  
    int sup; /* matricula do superior imediato */  
};  
typedef struct funcionario Funcionario;
```

- (a) Escreva uma função em C que recebe como parâmetros um vetor **contendo estruturas do tipo Funcionario**, através do ponteiro **vet**, para seu primeiro elemento, e do inteiro **n**, indicando seu número de elementos, e um inteiro **mat**, representando a matrícula de um funcionário. A função retorna o índice no vetor do funcionário correspondente, ou -1, se este não for encontrado. Protótipo:

```
int Indice_funcionario(Funcionario* vet, int n, int mat);
```

- (b) Escreva uma função em C que recebe como parâmetros **um vetor de ponteiros** para **Funcionario** (através do ponteiro **vet**, para seu primeiro elemento), um inteiro **n**, indicando seu número de elementos e um inteiro **mat**, representando a matrícula de um funcionário. A função deve retornar um novo vetor de estruturas **Funcionario** alocado dinamicamente contendo os funcionários que são subordinados diretos do funcionário cuja matrícula foi recebida por parâmetro. Um funcionário no topo da hierarquia, isto é, que não tem um superior imediato, tem o valor -1 no campo **sup**. Por exemplo, suponha que o vetor aponta para as funcionárias Luiza (**mat** = 123 e **sup** = 125), Diana (**mat** = 124 e **sup** = 129), Gina (**mat** = 125 e **sup** = 126), Dolores (**mat** = 126 e **sup** = -1), Beatriz (**mat** = 128 e **sup** = 126) e Ana (**mat** = 129 e **sup** = 128). Neste cenário, se a matrícula 126 for recebida por parâmetro, a função deve retornar um vetor de estruturas contendo os dados das funcionárias Gina e Beatriz. Protótipo:

```
Funcionario* determinaSubordinado(Funcionario** vet, int n, int mat);
```

Complementares

1. (*) Faça um programa que chame uma função que recebe por parâmetro os valores necessários para o cálculo da fórmula de báskara e retorna as suas raízes, caso seja possível calcular. Caso não seja possível, a função deve retornar $\{-1, -1\}$. Além disso, após receber os valores de retorno, a função `main` deve imprimir na tela uma mensagem de erro, se não foi possível calcular as raízes, ou as raízes, caso contrário (não use vetores para retorno).
2. (***) Para o controle dos veículos de uma cidade, a secretaria dos transportes criou o seguinte registro:

```
Proprietário:_____ Combustível:_____
Modelo:_____ Cor:_____
No.Chassi:_____ Ano:_____ Placa:_____
```

Sendo que o combustível pode ser álcool, gasolina ou diesel e a placa é formada por 3 letras e 4 dígitos. Sabendo que o número máximo de veículos da cidade é 5000, faça um algoritmo que:

- (a) Tenha um procedimento que leia os dados dos veículos até que seja fornecido um nome de proprietário em branco;
 - (b) Tenha um procedimento que imprima uma relação dos carros que sejam de um determinado ano em diante e sejam movidos com um determinado tipo de combustível;
 - (c) Tenha um procedimento que imprima uma relação com todos os dados da ficha, para os veículos cuja placa começa com uma determinada letra e termina com um determinado dígito;
 - (d) Tenha um procedimento que permita a troca de proprietário de um veículo, com o fornecimento do número do chassi.
3. (****) Considere o seguinte esqueleto de um código fonte C:

```
typedef float prova;
typedef float media;

struct provas{
    prova p[3];
    media M;
};
typedef struct provas Provas;

struct aluno{
    char ra[7];
    int frequencia;
    Provas Ps;
};

int main(void) {
    int n,i,j;
    float desvios_Ps[3]={0.0, 0.0, 0.0}, medias_Ps[3]={0.0, 0.0, 0.0};
    struct aluno alunos[MAX_ALUNOS];
    scanf("%d", &n);
    /* COMPLETE AQUI */
}
```

Complete o esqueleto acima. O programa deve ler um número inteiro n que indicará a quantidade de alunos na turma. Após isso, deve ler as notas das provas de cada aluno, calcular e mostrar na tela as notas e a média de cada aluno, a média geral de cada prova de todos os alunos e o desvio padrão de cada prova de todos os alunos. Note que não é necessária a declaração de mais nenhuma variável. Exemplo:

ENTRADA:

2

030034

4 5 6

123456

0 5 10

987654

6 9 3

SAIDA:

RA: 030034

Prova 0: 4.000000

Prova 1: 5.000000

Prova 2: 6.000000

Média: 5.000000

RA: 987654

Prova 0: 6.000000

Prova 1: 9.000000

Prova 2: 3.000000

Média: 6.000000

Média geral P0: 3.333333

Desvio padrão P0: 2.494438

Média geral P1: 6.333333

Desvio padrão P1: 1.885618

Média geral P2: 6.333333

Desvio padrão P2: 2.867442

RA: 123456

Prova 0: 0.000000

Prova 1: 5.000000

Prova 2: 10.000000

Média: 5.000000
