

Instituto Federal de Educação, Ciência e Tecnologia de Alagoas
Coordenadoria de Informática – Disciplina: algoritmos
Exercícios sobre Estruturas (registros) e Funções
Professora: Cledja Rolim

1- Seja a seguinte estrutura:

```
struct time {  
    int hora;  
    int min;  
    int seg;  
}
```

- a) Fazer um programa que recebe uma quantidade de tempo em segundos; declare uma variável do tipo *struct time*, e converta os segundos lidos em: horas, minutos e segundos, fazendo a inicialização da variável declarada.
- b) Ler a variável do tipo tempo (campos hora, min e seg) e devolver a quantidade de horas em segundos.
- c) Crie duas funções: uma que realize o que foi definido em (a) e outra função que realize (b); a função principal deve chamar as duas funções acima.

2- Seja uma estrutura para descrever os carros de uma determinada revendedora, contendo os seguintes campos:

```
marca: string de tamanho 15  
ano: inteiro  
cor: string de tamanho 10  
preço: real
```

- a) Escrever a definição da estrutura carro.
- b) Declarar o vetor *vetcarros* do tipo da estrutura definida acima, de tamanho 20.
- c) Definir uma função para ler o vetor *vetcarros*.
- d) Definir uma função que receba um preço e imprima os carros (marca, cor e ano) que tenham preço igual ou menor ao preço recebido.
- e) Defina uma função que leia a marca de um carro e imprima as informações de todos os carros dessa marca (preço, ano e cor).
- f) Defina uma função que leia uma marca, ano e cor e informe se existe ou não um carro com essas características. Se existir, informar o preço.
- g) Implemente uma função principal com um menu, para que o usuário possa selecionar uma das funções acima.

3- Seja uma estrutura contendo dados dos funcionários de uma empresa. Para cada funcionário tem-se os seguintes dados:

```
nome: string de tamanho 20  
CPF: vetor de 11 inteiros  
numpeças: inteiro  
salário: real
```

Implemente funções para cadastro, consulta e atualização e exclusão de funcionários.

4. Implemente uma função que recebe por parâmetro o raio de uma esfera e calcula o seu volume ($v = 4/3 \cdot P \cdot R^3$).
5. Implemente uma função que recebe as 3 notas de um aluno por parâmetro e uma letra. Se a letra for A a função calcula a média aritmética das notas do aluno, se for

P, a sua média ponderada (pesos: 5, 3 e 2) e se for H, a sua média harmônica. A média calculada também deve retornar por parâmetro.

6. Implemente uma função que recebe por parâmetro um valor inteiro e positivo e retorna o valor lógico Verdadeiro caso o valor seja primo e Falso em caso contrário.
7. Implemente uma função que recebe por parâmetro os valores necessário para o cálculo da fórmula de *baskara* e retorna, também por parâmetro, as suas raízes, caso seja possível calcular.
8. Implemente uma função que recebe por parâmetro o tempo de duração de uma fábrica expressa em segundos e retorna também por parâmetro esse tempo em horas, minutos e segundos.
9. Implemente uma função que recebe a idade de uma pessoa em anos, meses e dias e retorna essa idade expressa em dias.
10. Implemente uma função que verifique se um valor é perfeito ou não. Um valor é dito perfeito quando ele é igual a soma dos seus divisores excetuando ele próprio. (Ex: 6 é perfeito, $6 = 1 + 2 + 3$, que são seus divisores). A função deve retornar um valor booleano.
11. Implemente uma função que receba um valor inteiro e positivo e calcula o seu fatorial.
12. Implemente uma função que recebe por parâmetro um valor inteiro e positivo N e retorna o valor de S.
$$S = 1 + 1/1! + 1/2! + 1/3! + \dots + 1/N!$$
13. Estude o seguinte programa e aponte o valor de cada variável sempre que solicitado:

```
#include <stdio.h>
int num;
int func(int a, int b)
{
    a = (a+b)/2; /* Qual e o valor de a apos a atribuição? */
    num -= a;
    return a;
}

main()
{
    int first = 0, sec = 50;
    num = 10;
    num += func(first, sec); /* Qual e o valor de num, first e sec antes e depois da atribuição? */
    printf("\n\nConfira! num = %d\tfirst = %d\tsec = %d", num, first, sec);
}
```