

Algoritmos e Programação de Computadores

Atividades a serem desenvolvidas nas sessões de Laboratório

Sessão 11:

Objetivos:

- Exercitar a elaboração de programas que utilizam **estruturas de dados** do tipo:
 - **registro;**
 - **vetor de registros.**

1. Criar o programa abaixo:

```
#include <stdio.h>

/*****
* Descricao : ler e imprimir os dados de UM registro.
* Entrada: nome, idade, salario
* Saida:  nome, idade, salario
*****/

typedef struct {
    int idade;
    float salario;
    char nome[30];
} tipoDadosFunc;

int main() {
    tipoDadosFunc dadosfunc;
    int i;

    printf("Informe o nome: ");
    scanf("%s", dadosfunc.nome);
    printf("Informe a idade: ");
    scanf("%d", &dadosfunc.idade);
    printf("Informe o salario: ");
    scanf("%f", &dadosfunc.salario);
    printf("\n\nDADOS INFORMADOS");
    printf("\nNome: %s", dadosfunc.nome);
    printf("\nCod: %d", dadosfunc.idade);
    printf("\nSexo: %.2f", dadosfunc.salario);
    getchar();
    getchar();
}
```

- a) Corrija todos os erros sintáticos, se houver.
- b) Execute o programa com os dados que desejar e observe a saída:

2. Criar o programa abaixo:

```
#include <stdio.h>

/*****
* Descricao : ler e imprimir os dados de DEZ registros.
* Entrada: nome, idade, salario
* Saida:     nome, idade, salario
*****/

typedef struct {
    int idade;
    float salario;
    char nome[30];
} tipoDadosFunc;

int main() {
    tipoDadosFunc dadosfunc[10];
    int i;

    for (i = 0; i < 10; i++ ) {
        printf("Informe o nome: ");
        scanf("%s", dadosfunc[i].nome);
        printf("Informe a idade: ");
        scanf("%d", &dadosfunc[i].idade);
        printf("Informe o salario: ");
        scanf("%f", &dadosfunc[i].salario);
    }
    printf("\n\nDADOS INFORMADOS");
    for (i = 0; i < 10; i++ ) {
        printf("\n\nNome: %s", dadosfunc[i].nome);
        printf("\nCod: %d", dadosfunc[i].idade);
        printf("\nSalario: %.2f", dadosfunc[i].salario);
    }
    getchar();
    getchar();
}
```

- a) Corrija todos os erros sintáticos, se houver.
- b) Execute o programa com as informações sobre dez pessoas e observe a saída.

Nos programas abaixo, utilize como unidades de entrada e saída o teclado e o monitor, respectivamente, e utilize estruturas de dados do tipo vetor de registros.

3. Criar o programa abaixo:

```
#include <stdio.h>

/*****
* Descricao : ler e imprimir os dados de n registros.
* n deve ser lido e ser menor do que 10.
* O Programa principal somente chama funcoes para ler
* e mostrar os dados
* Entrada: nome, codigo, sexo, cargo
* Saida:  nome, codigo, sexo, cargo
*****/

typedef struct {
    int codigo;
    float salario;
    char nome[50], sexo[10], endereco[50], cargo[50];
} tipoDadosDeFuncionario;

void ledados (tipoDadosDeFuncionario *vetfunc, int *n) {
    int i;
    printf("Informe o nro de funcionarios: ");
    scanf ("%d",n);
    while ((*n<1) || (*n>10)) {
        printf("\nERRO! O nro de funcionarios deve ser entre 1 e 10.");
        printf("\nTecle <enter> para continuar.");
        getchar(); getchar();
        printf("\nInforme o nro de funcionarios: ");
        scanf ("%d",n);
    }
    for (i = 0; i<(*n); i++ ) {
        printf("\nInforme o nome: ");
        scanf("%s", vetfunc[i].nome);
        printf("Informe o cod: ");
        scanf("%d", &vetfunc[i].codigo);
        printf("Informe o sexo: ");
        scanf("%s", vetfunc[i].sexo);
        printf("Informe o cargo: ");
        scanf("%s", vetfunc[i].cargo);
    }
}

void mostrados (tipoDadosDeFuncionario *vetfunc, int n) {
    int i;
    for (i = 0; i<n; i++ ) {
        printf("\n\nNome: %s",vetfunc[i].nome);
        printf("\nCod: %d",vetfunc[i].codigo);
        printf("\nSexo: %s",vetfunc[i].sexo);
        printf("\nCargo: %s",vetfunc[i].cargo);
    }
}
```

```

int main() {
    tipoDadosDeFuncionario dadosDeFunc[10];
    int i,nrofunc;

    ledados(dadosDeFunc,&nrofunc);
    mostrados(dadosDeFunc,nrofunc);
    getchar();
    getchar();
}

```

4. Faça um programa que leia e guarde numa tabela códigos e nomes de profissões. Dado o código (inteiro) de uma determinada profissão, encontre e imprima o respectivo nome (20 posições). O usuário deve informar o tamanho da tabela no início do programa.
5. Faça um programa que leia os dados (nome, cargo, departamento, salário, data de admissão) de 10 funcionários e imprima a lista dos empregados que ganham salário maior que a média geral.
6. Faça um programa que leia os dados de uma agenda telefônica com a estrutura abaixo e imprima os contatos que fazem aniversário no mês de dezembro. A agenda deve conter 10 registros.

Agenda

```

Nome
Endereço
Telefone
Aniversário
    Dia
    Mês
    Ano

```

7. Altere o programa do item anterior para que mostre os contatos em ordem alfabética de nome. **Crie uma função para ordenar o vetor de registros.** Na função de ordenação, você precisará comparar dois nomes para verificar se deverá trocar os registros de posição ou não. Essa comparação pode ser feita utilizando a função `strcmp()` da biblioteca `string.h`, da seguinte forma:

```

If strcmp(registro[i].nome, registro[i+1].nome) > 0 {
    /* deve trocar os registros, pois nome do registro na posição i
    é lexicograficamente maior do que nome do registro na posição
    i+1*/
}

```

obs1: a função `strcmp(s1,s2)` retorna um número negativo (<0) se `s1` for lexicograficamente menor do que `s2`, zero se forem iguais, ou um número positivo (>0) se `s1` for lexicograficamente maior do que `s2`.

obs2: lembre-se de que você pode trocar a posição do registro inteiro de uma vez só. Não é necessário fazer campo a campo do registro.