

Aula 13 - 02/05/17

1) Registros (cont.)

Exemplo: Preencher e imprimir um vetor de registros.

```
#include <stdio.h>
#include <string.h>
#define TAM 50
#define MAX_STRING 100

typedef struct Pessoa {
    char nome[MAX_STRING];
    int idade;
} TPessoa;

void preenche_vetor(TPessoa pessoas[], int num_pessoas) {
    int i;
    for (i=0; i<num_pessoas; i++) {
        printf("Entre com o nome:\n");
        fgets(pessoas[i].nome, MAX_STRING-1, stdin);
        printf("Entre com a idade:\n");
        scanf("%d", &(pessoas[i].idade));
        getchar();
    }
}

void imprime_vetor(TPessoa pessoas[], int num_pessoas) {
    int i;
    for (i=0; i<num_pessoas; i++) {
        printf("\nDados da pessoa %d:\n", i+1);
        printf("nome = %s", pessoas[i].nome);
        printf("idade = %d\n", pessoas[i].idade);
    }
}

int main(void) {
    TPessoa pessoas[TAM];
    int num_pessoas;

    printf("Entre com o número de pessoas:\n");
    scanf("%d", &num_pessoas);
    getchar();

    preenche_vetor(pessoas, num_pessoas);
    imprime_vetor(pessoas, num_pessoas);

    return 0;
}
```

Código 1. Programa que preenche e imprime um vetor de registros.

2) Exercícios

- Implementar e testar o Código 1.

3) Atividade Avaliativa 2

1. Faça um programa em C para realizar o cadastro de estudantes. O programa deve realizar as seguintes operações:

- **Cadastrar** um estudante, contendo os dados ID, nome, data de nascimento e CPF.
- **Listar** todos os estudantes cadastrados. Essa operação deve exibir os dados de cada estudante.
- **Pesquisar** um estudante pelo nome. Essa operação deve exibir os dados do estudante.
- **Atualizar** um estudante pelo ID. Todos os dados do estudante devem ser atualizados, exceto o ID.
- **Remover** um estudante pelo ID.

O programa deve permitir a realização iterativa das operações, enquanto não for digitado 6 para sair (Figura 1). Deverão ser exibidas mensagens informando o estado das operações, por exemplo: “Cadastro realizado com sucesso”, “Estudante não localizado”, entre outras.

Cadastro de Pessoas

1 – Cadastrar
2 – Listar
3 – Pesquisar
4 – Atualizar
5 – Remover
6 - Sair

Figura 1. Opções do cadastro de pessoas.

Observações

- (1) A atividade deve ser realizada em grupo (grupo do projeto integrador).
- (2) A nota da atividade está condicionada à apresentação do programa.
- (3) Faça os programas seguindo as orientações abaixo.

Orientações

- Defina o número máximo de estudantes que podem ser cadastrados.
- Defina um tipo de dados para representar um estudante, contendo os campos id, nome, data de nascimento e CPF. Adicionalmente, crie um campo chamado situação.
- O campo situação deve ser utilizado para guardar a situação de um cadastro (1 – ativo; 0 – inativo). Assim, ao cadastrar um estudante, o campo situação deve ser igual a 1. Quando o cadastro for removido, esse campo deve ser zerado, indicando que o cadastro está inativo.
- Use um vetor de registros para armazenar os cadastros.

- Use uma variável apontadora para indexar as posições do vetor de estudantes. Essa variável deve sempre apontar para a próxima posição disponível do vetor, no qual será inserido o cadastro do estudante. No início, essa variável é inicializada com 0 (zero). Após a inserção de um cadastro, essa variável deve ser incrementada em 1 (um), informando a próxima posição livre no vetor.
- Faça uma função para realizar a operação *cadastrar*. Essa função deve receber como parâmetros o vetor de estudantes e a posição de inserção (variável apontadora). A função deve realizar a leitura dos dados do estudante e armazenar esses dados no vetor na posição de inserção. Resumindo, essa função realiza o cadastro de UM estudante apenas.
- Nessa operação, não solicite o ID para o usuário, apenas o nome, data de nascimento e CPF. Atribua como ID o valor da variável apontadora. Assim, o ID do estudante será igual a sua posição no vetor de estudantes. Lembre-se que o campo situação deve ser igual a 1 (um).
- Faça uma função para realizar a operação *listar*. Essa função deve receber como parâmetros o vetor de estudantes e o número de pessoas cadastradas até o momento (variável apontadora, pois o seu valor é igual ao número de pessoas cadastradas). Assim, essa função deve percorrer todas as posições do vetor e imprimir cada cadastro. Lembre-se que se deve listar apenas cadastros que estão ativos.
- Faça uma função para realizar a operação *pesquisar*. Essa função deve receber como parâmetros o vetor de estudantes e o número de pessoas cadastradas até o momento (variável apontadora). Dentro da função, solicite um nome para o usuário (uma string). Após, percorra todas as posições do vetor, verificando se o nome do estudante atual é igual ao nome informado. Se o estudante for localizado, imprima o cadastro correspondente. Lembre-se que se deve pesquisar apenas por estudantes cujo cadastro está ativo.
- Faça uma função para realizar a operação *atualizar*. Essa função deve receber como parâmetro o vetor de estudantes. Dentro da função, solicite o ID para o usuário (um número inteiro). Sabe-se que o ID corresponde à posição do vetor na qual o cadastro está armazenado. Assim, solicite os dados do estudante para o usuário, e sobrescreva os dados armazenados anteriormente na posição indicada. Lembre-se que o ID não deve ser alterado.
- Faça uma função para realizar a operação *remover*. Essa função deve receber como parâmetro o vetor de estudantes. Dentro da função, solicite o ID para o usuário (um número inteiro). Sabe-se que o ID corresponde à posição do vetor na qual o cadastro está armazenado. Assim, a operação remover consiste apenas em zerar o campo situação do cadastro armazenado na posição indicada.

- Por fim, defina o critério de parada do laço principal. O programa deve terminar quando o usuário digitar a opção 6 ou se a variável contadora ultrapassar o número máximo de posições definido.
- Cuidado com os limites do vetor. Caso a variável apontadora tenha valor igual ou maior que o número máximo de posições definido, a função que realiza a operação *cadastrar* não deve ser chamada. Quando esse fato ocorrer, o programa deve ser finalizado.