

Prática 7.2

Introdução à programação C/C++
2017/2

Estruturas - parte 2
Universidade Federal do Rio de Janeiro
Prof. Eduardo Mangeli

Exercícios

Exercício 1. Considere a seguinte estrutura:

```
typedef struct {  
    int identificador;  
    char nome[42];  
} INDIVIDUO;
```

Escreva um programa que pergunte ao usuário quantos indivíduos serão informados e use alocação dinâmica para reservar o espaço necessário para armazená-los na memória. No caso de a alocação de memória falhar, o programa deve emitir a mensagem de erro 'Nao ha memoria suficiente', e terminar.

Atenção! Use apenas notação de ponteiros (evite a notação de vetores) para armazenar os valores dos indivíduos.

Exercício 2. Modifique o programa do Exercício 1 e inclua nele a função `preenche` que deve receber um ponteiro para `INDIVIDUO`, o seu tamanho, e ter as seguintes funcionalidades:

- i) solicitar a entrada do nome de cada um dos indivíduos (deve ser permitida a entrada de nomes com espaços)
- ii) criar, para cada um dos indivíduos, um identificador pseudo aleatório entre 1 e 999

Mantenha o seguinte protótipo para a função `preenche`:

```
void preenche(INDIVIDUO * dados, int tam)
```

função `preenche`

Sua função deve modificar (preencher) o a região da memória apontada pelo ponteiro recebido como entrada. Ela deve ser chamada se a alocação de memória realizada pelo programa ocorrer de maneira adequada.

Atenção! Use apenas notação de ponteiros (evite a notação de vetores).

Exercício 3. Modifique o programa iniciado no Exercício 1 e inclua nele a função `imprime` que deve receber um ponteiro para `INDIVIDUO`, o seu tamanho, e imprimir os os nomes e identificadores de cada um dos indivíduos.

Mantenha o seguinte protótipo para a função `imprime`:

```
void imprime(INDIVIDUO * individuos, int quantos)
```

função `imprime`

Sua nova função deve ser chamada após o preenchimento dos dados dos indivíduos realizado com a função do Exercício 2.

Atenção! Use apenas notação de ponteiros (evite a notação de vetores).

Exercício 4. Modifique o programa iniciado no Exercício 1 e inclua nele a função `maiorIdentificador` que deve receber um ponteiro para `INDIVIDUO`, o seu tamanho, e retornar um ponteiro para o `INDIVIDUO` com o maior identificador.

Mantenha o seguinte protótipo para a função `maiorIdentificador`:

```
void maiorIdentificador(INDIVIDUO * dados, int tam)
```

função `maiorIdentificador`

Após a impressão dos dados realizada pela função desenvolvida no Exercício 3, use o ponteiro retornado pela função `maiorIdentificador` para imprimir o nome do INDIVIDUO como maior identificador.

Exercício 5. Faça duas modificações no programa desenvolvido até agora:

- Use o ponteiro retornado pela função desenvolvida no Exercício 4 para alterar o valor do maior identificador para 0 (zero).
- Use a função `imprime` para imprimir os dados de todos os INDIVIDUOS depois de alterar o valor do maior identificador

A Listagem 1 mostra um exemplo de funcionamento do programa.

Exercício 6. Modifique as funções `imprime`, `preenche` e `maiorIdentificador` para que, ao invés de usar a notação de ponteiros, usem apenas a notação de vetores.

```
Quantos individuos serao informados?
3
Entre com o 1o nome: joao carlos
Entre com o 2o nome: andre felipe
Entre com o 3o nome: mariano

Individuos
=====
Nome: joao carlos
Identificador: 332
-----
Nome: andre felipe
Identificador: 149
-----
Nome: mariano
Identificador: 811
=====

Maior identificador: mariano

Individuos
=====
Nome: joao carlos
Identificador: 332
-----
Nome: andre felipe
Identificador: 149
-----
Nome: mariano
Identificador: 0
=====
```

Listagem 1: exemplo de funcionamento do programa, os identificadores são pseudo aleatórios