

## PROCESSO SELETIVO - 05 / 2023

### Área de Conhecimento: Linguagem de Programação

#### PROVA ESCRITA - PADRÃO DE RESPOSTA

**QUESTÃO 1:** Dados dois inteiros,  $m$  e  $n$  lidos da entrada padrão, escreva um programa na Linguagem C, que imprima, na saída padrão, o MMC (mínimo múltiplo comum) entre  $m$  e  $n$ .

Por exemplo, para  $m = 12$  e  $n = 20$ , o programa deve imprimir 60.

RESPOSTA:

Primeira solução possível:

```
//  
// Função mmc  
//  
int mmc(int num1, int num2) {  
  
    int resto, a, b;  
  
    a = num1;  
    b = num2;  
  
    do {  
        resto = a % b;  
  
        a = b;  
        b = resto;  
  
    } while (resto != 0);  
  
    return ( num1 * num2) / a;  
}
```

Segunda solução possível:

```
//  
// Função mmc  
//  
int mmc(int num1, int num2) {  
  
    int mmc, aux, i;  
  
    for (i = 2; i <= num2; i++) {  
        aux = num1 * i;  
        if ((aux % num2) == 0) {  
            mmc = aux;  
            i = num2 + 1;  
        }  
    }  
    return mmc;  
}
```

Bibliografia: DEITEL, P. DEITEL, H. C: como programar. 6a edição. São Paulo. Pearson Prentice Hall,2011. (Capítulo 5, Exercício 5.29)

TENENBAUM, A.M.; LANGSAM, Y.; AUGENSTEIN, M.J.Estruturas de Dados Usando C. SãoPaulo: Makron Books, 1995. ISBN 85-346-0348-0 (Capítulo 3, Exercício 3.2.2)

### **Membros da Banca:**

---

**Gilmário Barbosa dos Santos**

---

**Janine Kniess**

---

**Cristiano Damiani Vasconcellos**

---

**André Tavares da Silva (Presidente)**

## PROCESSO SELETIVO - 005 / 2022

### Área de Conhecimento: Linguagem de Programação

#### PROVA ESCRITA - PADRÃO DE RESPOSTA

**QUESTÃO 2:** Existem geralmente duas formas de passagem de parâmetro em funções: por valor e por referência (ALGOL usa passagem por nome). Na linguagem C, todas as chamadas são feitas por valor, mas é possível simular a chamada por referência usando operadores de endereço e de indireção. Dependendo do que está sendo passado por parâmetro, justifique qual das maneiras é a mais adequada para cada situação (exemplo: tipo primitivo, estrutura definida pelo usuário, função, matriz, etc.). Explique como pode ser codificada a passagem de parâmetro por referência na linguagem C.

RESPOSTA:

Em C, você usa ponteiros e operadores de indireção para simular uma chamada por referência. Ao chamar uma função com argumentos que devem ser modificados, os endereços dos argumentos são passados. Isso normalmente é feito ao se aplicar o operador (&) à variável (na função chamadora), cujo valor será modificado. ~~Como vimos no Capítulo 6,~~ os arrays não são passados usando-se o operador &, pois C passa automaticamente o local inicial para a memória do array (o nome de um array é equivalente a &arrayName[0]). Quando o endereço de uma variável é passado para uma função, o operador de indireção (\*) pode ser usado na função para modificar o valor nesse local da memória da função chamadora.

Bibliografia:

DEITEL, P. DEITEL, H. C: como programar. 6a edição. São Paulo. Pearson Prentice Hall, 2011. (Seção 7.4 *Passando argumentos para funções por referência*)

SCHILDT, H. C completo e total. São Paulo. Makron Books, McGraw-Hill, 1996. (Capítulo 6)

#### Membros da Banca:

---

Gilmário Barbosa dos Santos

---

Janine Kniess

---

Cristiano Damiani Vasconcellos

---

André Tavares da Silva (Presidente)

## PROCESSO SELETIVO - 005 / 2022

### Área de Conhecimento: Linguagem de Programação

#### PROVA ESCRITA - PADRÃO DE RESPOSTA

**QUESTÃO 3:** Faça uma função recursiva, em linguagem C, que calcule o valor da série **S** descrita abaixo para um valor inteiro  $n > 0$  a ser fornecido como parâmetro para a mesma (retorna zero caso  $n \leq 0$ ).

$$S = 1 + 1/1! + 1/2! + 1/3! + \dots + 1/n!$$

RESPOSTA:

```
//  
// Função fatorial  
//  
int fatorial(int n) {  
    if(n<=0) return 1;  
    return n * fatorial(n-1);  
}  
  
//  
// Função S = 1 + 1/1! + 1/2! + 1/3! + ... + 1/n!  
//  
float S(int n) {  
    if(n=0) return 1;  
    return 1.0/fatorial(n) + S(n-1);  
}
```

Bibliografia:

MANZANO, José Augusto N. G; OLIVEIRA, Jayr Figueiredo de Algoritmos: lógica para desenvolvimento de programação de computadores. 27. ed. rev. São Paulo: Érica, 2014. (Seção 10.7 Funções e Recursividade)

DEITEL, P. DEITEL, H. C: como programar. 6a edição. São Paulo. Pearson Prentice Hall, 2011. (Seção 5.14 Recursão)

SCHILDT, H. C completo e total. São Paulo. Makron Books, McGraw-Hill, 1996. (Capítulo 6)

#### Membros da Banca:

---

Gilmário Barbosa dos Santos

---

Janine Kniess

---

Cristiano Damiani Vasconcellos

---

André Tavares da Silva (Presidente)

## PROCESSO SELETIVO - 005 / 2022

### Área de Conhecimento: Linguagem de Programação

#### PROVA ESCRITA - PADRÃO DE RESPOSTA

**QUESTÃO 4:** Suponha que o tipo de nó de uma Árvore Binária seja definido assim:

```
struct nodeType{
    int valor;
    struct nodeType *left, *right;
};
```

Caminhamento em árvore é o ato de percorrer todos os nós de uma forma sistemática sendo que cada nó é visitado uma única vez. Um caminhamento completo sobre uma árvore produz uma sequência linear dos nós, de modo que cada nó da árvore passa a ter um nó seguinte ou um nó anterior, ou ambos, para uma dada forma de caminhamento. No caso de árvores binárias existem 3 tipos de caminhamento mais frequentemente utilizados: Caminhamento Pós-Ordem (LRN); Caminhamento Pré-Ordem (NLR) e Caminhamento Em-Ordem (LNR). Onde: L = Left, R = Right e N = Node.

Escreva uma função para exibir na saída padrão o um dos caminhamentos à sua escolha.

RESPOSTA:

Solução possível - Pós-Ordem:

```
//
// Função PosOrdem
//
void PosOrdem(struct nodeType *node) {
    if(node) {
        PosOrdem(node->left);
        PosOrdem(node->right);
        printf("%d ", node->valor);
    }
}
```

Solução possível - Pré-Ordem:

```
//  
// Função PreOrdem  
//  
void PreOrdem(struct nodeType *node) {  
    if(node) {  
        printf("%d ", node->valor);  
        PreOrdem(node->left);  
        PreOrdem(node->right);  
    }  
}
```

Solução possível - Em-Ordem:

```
//  
// Função EmOrdem  
//  
void EmOrdem(struct nodeType *node) {  
    if(node) {  
        EmOrdem(node->left);  
        printf("%d ", node->valor);  
        EmOrdem(node->right);  
    }  
}
```

Bibliografia:

DEITEL, P. DEITEL, H. C: como programar. 6a edição. São Paulo. Pearson Prentice Hall,2011. (Seção 12.7 *Arvores*)

TENENBAUM, A.M.; LANGSAM, Y.; AUGENSTEIN, M.J.Estruturas de Dados Usando C. SãoPaulo: Makron Books, 1995. ISBN 85-346-0348-0 (Seção 5.1 *Árvores Binárias*)

#### **Membros da Banca:**

---

**Gilmário Barbosa dos Santos**

---

**Janine Kniess**

---

**Cristiano Damiani Vasconcellos**

---

**André Tavares da Silva (Presidente)**



# Assinaturas do documento



Código para verificação: **6T12XLB9**

Este documento foi assinado digitalmente pelos seguintes signatários nas datas indicadas:



**ANDRE TAVARES DA SILVA** (CPF: 908.XXX.020-XX) em 25/09/2023 às 09:04:07

Emitido por: "SGP-e", emitido em 30/03/2018 - 12:44:17 e válido até 30/03/2118 - 12:44:17.

(Assinatura do sistema)



**JANINE KNISS** (CPF: 887.XXX.119-XX) em 25/09/2023 às 11:44:12

Emitido por: "SGP-e", emitido em 30/03/2018 - 12:36:26 e válido até 30/03/2118 - 12:36:26.

(Assinatura do sistema)



**GILMARIO BARBOSA DOS SANTOS** (CPF: 410.XXX.285-XX) em 25/09/2023 às 13:01:58

Emitido por: "SGP-e", emitido em 30/03/2018 - 12:35:07 e válido até 30/03/2118 - 12:35:07.

(Assinatura do sistema)

Para verificar a autenticidade desta cópia, acesse o link <https://portal.sgpe.sea.sc.gov.br/portal-externo/conferencia-documento/VURFU0NfMTlwMjJfMDAwNDE4NjZfNDE5MDhfMjAyM182VDEyWEExCOQ==> ou o site <https://portal.sgpe.sea.sc.gov.br/portal-externo> e informe o processo **UDESC 00041866/2023** e o código **6T12XLB9** ou aponte a câmera para o QR Code presente nesta página para realizar a conferência.