

## Lista de Exercícios 4

### Instruções

A) As questões são individuais. Em caso de cópias de trabalho a pontuação será zero para os autores originais e copiadores. Não serão aceitas justificativas como: “Fizemos o trabalho juntos, por isso estão idênticos”.

B) Para cada exercício que solicite a construção de função/procedimento procure responder antes de programar as seguintes perguntas:

- a) O objetivo do procedimento/função
- b) O nome do procedimento/função
- c) Os parâmetros de entrada
- d) O tipo de dado da saída, caso seja uma função.

C) Após ter clareza sobre os itens acima, documente cada uma das funções/procedimentos por meio comentários (obrigatório).

A seguir um exemplo de uma função que recebe um número  $n$  por parâmetro calcula e retorna o seu cubo de  $n$ .

/\*

Descrição: a função tem por objetivo calcula o cubo de  $n$ .

Entrada:  $n$  (inteiro)

Saída: inteiro

\*/

```
int cubo(int n) {  
    return n*n*n;  
}
```

### Parte A - Questões para serem entregues no Canvas

1. Construa uma função que receba parâmetro duas datas, formada por três inteiros cada (dia, mes, ano). Calcule e retorne o número de dias entre essas duas datas. Fique à vontade para ignorar o impacto dos anos bissextos e também para considerar todos os meses com 30 dias.

Main: a função *main* deverá ler da tela as datas no formato d/m/a e posteriormente invocar a função criada passando as datas. Feito isso, o algoritmo deverá exibir na tela um número inteiro.

2. Construa um algoritmo que realize a soma de dois números reais. Nele, construa as seguintes funções:

- (a) um procedimento que se encarrega de apresentar o programa na tela;
- (b) uma função *float* que receba dois valores reais e retorne a soma dos dois parâmetros recebidos;
- (c) um procedimento que receba um valor real relativo ao resultado da soma e o escreva na tela;
- (d) uma função *main* que se encarregue de estabelecer o fluxo das chamadas das funções acima para atender ao propósito desta questão.

3. Observe o código abaixo.

```
...
int a=5,
b=2;
troca(&a, &b);
printf("\n%i, %i", a, b);
...

void troca(int *x, int *y){
    int aux;
    aux= *x;
    *x = *y;
    *y = aux;
}
```

a) Responda que valores serão escritos na tela após a execução.

b) Comente cada linha do programa explicando que é ela faz e quais serão os valores as variáveis envolvidas naquela linha.

4. Crie uma nova versão da questão que calcula a diferença entre duas datas. Modifique a função para um procedimento que implementa a leitura das datas pelo teclado. Substitua a estratégia de retornar o valor lido pela seguinte: o valor lido deverá ser armazenado na variável parametrizada por referência. Teste a sua solução a partir da implementação de um *main*.

5. Implemente um procedimento chamado *primo* que recebe como parâmetro um inteiro *m*, passado por valor, e dois outros inteiros, *p1* e *p2*, passados por referência. O procedimento deve armazenar em *p1* o **maior** número primo menor do que *m* e deve armazenar em *p2* o **menor** número primo maior do que *m*.

Implemente também uma função *main* que solicita ao usuário o valor de *m*, chama o procedimento primo e depois imprime o resultado.

6. Escreva um procedimento chamado teste que recebe um valor *n* passado “por valor” e dois inteiros *b* e *k* passados “por referência”. Seu procedimento deve armazenar em *b* e *k* valores tais que  $bk = n$  e *b* seja o menor valor possível.

Implemente também uma função *main* que solicita ao usuário o valor de *n*, chama o procedimento criado e depois imprime o resultado.

7. Crie um programa que receber três valores reais, calcule e determina o segundo maior valor. Utilize a estratégia de ordenar os três valores lidos. O seu programa deverá resolver esse problema de diferentes maneiras a serem implementadas:

- a) Um procedimento que recebe parâmetros por valor e mostra o resultado;
- b) Uma função que recebe parâmetros por valor e retorna o resultado para o main;
- c) Um procedimento que recebe os valores por parâmetro e retorna por referência o segundo maior para o main;

O programa deverá conter, além da função principal, as seguintes outras:

- d) Um procedimento para ordenar três valores reais (passagem de parâmetro por referência);
- e) Uma função para escrever o resultado (o segundo maior valor lido).

### **Parte B - Questões para treino sobre Funções e Procedimentos – Não precisa entregar**

1. Construa um programa que calcule a área de um retângulo. Planeje, com calma, cada uma das funções/procedimentos que deverão compor o programa:

- (a) Um procedimento para a apresentação do objetivo do software;
- (b) Um procedimento para a leitura de cada um dos dados necessários para solucionar o problema (base e altura), rejeitando a leitura de valores negativos;
- (c) Uma função para o cálculo da área;
- (d) Um procedimento para a escrita do resultado, e;
- (e) Uma função *main*, encarregada de orquestrar o fluxo das chamadas para atender ao propósito desta questão.

2. Construa um programa que permita calcular o peso ideal de uma pessoa. Planeje com calma as funções necessárias para atender ao propósito da questão:

- a) Um procedimento deverá fazer a apresentação na tela deste programa.
- b) Uma função deverá ler e retornar o gênero [M/F] (*char*). Rejeitar a leitura de caracteres fora deste domínio. A função deverá garantir que o caractere retornado esteja em maiúsculo, independentemente da forma que ele foi digitado pelo usuário do programa.
- c) Uma função deverá ler e retornar a idade (int). Rejeitar a leitura de valores negativos.
- d) Uma função deverá receber dois parâmetros, o gênero e a idade. A partir da interpretação destes dois parâmetros, gerar e retornar um valor relativo ao peso ideal.
- e) Um procedimento deverá receber um valor relativo ao resultado e escrever o peso ideal.
- f) Uma função *main*, encarregada de orquestrar o fluxo das chamadas para atender ao propósito desta questão.

3. Construa um programa para o problema dos triângulos. Ele deverá conter as seguintes funcionalidades a serem implementadas em funções/procedimentos:

a) Uma função que verifique se os valores apresentados como lados do triângulo atendem ou não à condição para construir triângulos.

- Argumentos da função: os três valores que representam cada lado do triângulo.

- Valor gerado: verdadeiro, se os três lados permitem a construção de um triângulo; ou falso, caso contrário.

b) Uma função que verifique o tipo do triângulo.

- Argumentos da função: os três lados de um triângulo.

- Valor gerado: 1, se equilátero; 2, se isóscele; 3, se escaleno; 0, caso não se trate de um triângulo (para este caso, evocar a função construída no item acima).

c) Uma função que leia através do teclado o valor de um lado de um triângulo.

- Argumento da função: Um número inteiro (1..3) que expresse qual o lado a ser lido, a fim de enriquecer a mensagem enviada à tela para o usuário;

- Valor gerado: a função deverá retornar o valor lido.

Obs: Consistir o dado lido (rejeitar a leitura de lado negativo)

d) Um procedimento que escreva na tela uma *string* relativa a um tipo do triângulo, a saber: “Equilátero”, “Isóscele” ou “Escaleno”.

- Argumento de entrada: Um valor inteiro relativo ao tipo do triângulo, a saber: 1 (representando equilátero), 2 (representando isóscele); 3 (representando escaleno).

- Valor gerado: nenhum (vazio).

Obs: Nenhuma mensagem deverá ser escrita se o valor parametrizado for diferentes de 1..3.

e) Um procedimento que faça uma apresentação ao usuário, com mensagens elucidativas sobre os objetivos do programa.

- Argumentos da função: Nenhum

- Valor gerado: Nenhum (vazio)

f) Uma função principal que gerencie o fluxo das chamadas às demais funções para atender ao propósito do problema tratado

### **Parte C - Questões para treino sobre Ponteiros – Não precisa entregar**

1. Implemente um programa em C que leia dois inteiros, armazenando-os em variáveis. O programa deve comparar os endereços das variáveis e exibir o maior deles.

2. Faça um procedimento que leia 3 números inteiro (a, b, c). Para cada valor lido, mostre o nome da variável, o endereço e o seu valor, conforme exemplificado a seguir:

NOME_VARIAVEL	ENDEREÇO	VALOR
a	0000fff0	5
b	0000fff4	9
c	0000fff8	1

Invoque o procedimento por meio do programa principal/*main*.

3. Explique cada uma das expressões a seguir, indicando a diferença entre elas:

- `p++;`
- `(*p)++;`
- `*(p++);`

Qual informação se refere a expressão `*(p+10)`?

4. Identifique o erro no programa a seguir, de modo que seja exibido o valor 10 na tela.

```
#include <stdio.h>
int main()
{
    int x, *p, **q;
    p = &x;
    q = &p;
    x = 10;
    printf("\n%d \n", &q);
    return(0);
}
```