

Lista de Exercícios 4

Informações sobre cópias

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”.

Documentação e comentários

Para os exercícios desta lista 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.

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 1 – Exercícios para serem entregues

Resolva os exercícios a seguir e entregue pelo CANVAS. Cada exercício deve conter um arquivo no **formato .C**. As funções/procedimentos devem ser documentados usando comentário (vide exemplo acima).

1. Escreva um **procedimento** que receba 3 notas de um aluno por parâmetro e uma letra. Se a letra for "A", o procedimento calcula e escreve a média aritmética das notas do aluno, se for "P", calcula e escreve a sua média ponderada (pesos: 5, 3 e 2). Escreva também um programa *main* que leia 3 notas de n alunos e acione o procedimento para cada aluno.

2. A prefeitura de uma cidade fez uma pesquisa entre os seus habitantes, coletando dados sobre o salário familiar e quantidade de membros (pais + filhos). Escreva um **procedimento** que leia esses dados para um número não determinado de famílias (ler até que salário igual 0), calcule e exiba a média de salário da população. Faça um programa *main* que acione o **procedimento**.

3. Escreva um **procedimento** que recebe 3 valores inteiros por parâmetro e os exiba em ordem crescente. Faça um programa *main* que leia do teclado n conjuntos de 3 valores e acione o procedimento para cada conjunto.

4. Escreva uma **função** que receba como parâmetro um valor n inteiro positivo, calcule e retorne o valor de E , definido pela equação abaixo

$$E = 1 + \frac{1}{1!} + \frac{1}{2!} + \frac{1}{3!} + \dots + \frac{1}{n!}$$

5. Escreva uma **função** que receba como parâmetro um valor n inteiro positivo, calcule e retorne o valor da soma S , definida pela equação abaixo:

$$S = \frac{2}{4} + \frac{5}{5} + \frac{10}{6} + \frac{17}{7} + \frac{26}{8} + \dots + \frac{n^2 + 1}{n + 3}$$

6. Escreva um **procedimento** que recebe por parâmetro 3 valores reais X , Y e Z e que verifique se esses valores podem ser os comprimentos dos lados de um triângulo e, neste caso, exibe qual é o tipo de triângulo formado. Para que X , Y e Z formem um triângulo é necessário que a seguinte propriedade seja satisfeita: o comprimento de cada lado de um triângulo é menor do que a soma do comprimento dos outros dois lados. O procedimento deve identificar o tipo de triângulo formado observando as seguintes definições:

1. Triângulo Equilátero: os comprimentos dos 3 lados são iguais;
2. Triângulo Isósceles: os comprimentos de pelo menos 2 lados são iguais;
3. Triângulo Escaleno: os comprimentos dos 3 lados são diferentes;

Faça um programa que leia um número indeterminado de triângulos (valores dos 3 lados) e para cada triângulo, acione o **procedimento**.

7. 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.

8. 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*.

9. 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.

10. 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.

Parte 2 – Exercícios Recomendados (não precisam ser entregues)

1. Escreva uma **função** que receba como parâmetros 3 números inteiros e retorne o menor valor. Imprima, na função *main*, o menor valor retornado.

2. Escreva um **procedimento** que recebe a média final de um aluno, identifica e exibe o seu conceito, conforme a tabela abaixo. Faça um programa que leia a média de *n* alunos, acionando o procedimento para cada um deles. O valor de *n* deve ser lido do teclado no *main*.

Nota	Conceito
até 39	F
40 a 59	E
60 a 69	D
70 a 79	C
80 a 89	B
à partir de 90	A

3. Escreva uma **função** que receba como parâmetro um valor n inteiro positivo, calcule e retorne o valor da soma S , definida pela equação abaixo:

$$S = 1 + \frac{1}{2} + \frac{1}{3} + \frac{1}{4} + \dots + \frac{1}{n}$$

4. Escreva uma **função** que lê um número indeterminado de notas de alunos, calcula e retorna a média das notas dos alunos aprovados (nota maior ou igual a 6). Faça um programa que leia o número de alunos e imprima a média retornada pela função.

5. Crie uma **função** que recebe três valores booleanos (A, B e C) e retorne o valor verdade da operação ($\sim A \vee B \wedge C$) em relação aos três valores. Crie um programa *main* que ilustre o uso da função. Utilize a biblioteca `stdbool` para trabalhar com valores booleanos.

6. Construa uma **função** que receba dois inteiros positivos (n e k), calcule e retorne o k -ésimo dígito da direita para a esquerda de um número n . Suponha que k é menor ou igual ao número de dígitos de n . Exemplo: A função recebe o número 1998 e o k desejado ($k=3$). Nesse exemplo, o algoritmo deverá retornar o 3º dígito de 1998 que corresponde a 9. Crie um programa *main* que ilustre o uso da função.

7. 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.

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

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

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

9. 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);
}
```