

# Quarta lista de exercícios de Introdução à Programação de Computadores

Para todos os exercícios desta lista:

- Elabore um conjunto significativo de casos de testes.
- Implemente o programa usando GCC no Ubuntu.
- Use o padrão ISO C 2011 (C11).
- Teste o programa usando os dados do item *a*.

## Exercício 1

1. Qual é o tipo de cada uma das constantes a seguir?

- a) `"7"`
- b) `3`
- c) `"3.14159"`
- d) `'2'`
- e) `5.1`
- f) `"Alo"`
- g) `15L`
- h) `8.5L`
- i) `8.5f`
- j) `'\a'`

2. Qual é a saída do fragmento de código abaixo? **Nota:** nesse tipo de exercício você deve pensar primeiro e só depois executar o código para confirmar a resposta.

```
int x = 10;
char w = 'Y';
float z = 5.1234

printf("\nPrimeiro\nExemplo\n:");
printf("%5d\n, w eh %c\n", x, w);
printf("\nz eh %8.2f\n", z);
```

3. Escreva um programa que peça ao usuário que digite um número inteiro e depois imprima o inteiro primeiro como um caracter, depois como um decimal e, finalmente, como um float. Um exemplo de execução é este:

```
O número como um caracter: K
O número como um decimal : 75
O número como um float : 0.000000
```

4. Escreva um programa que leia nove inteiros e imprima-os três a três na mesma linha separados por vírgulas, como mostrado abaixo.

Entrada:

```
10 31 2 73 24 65 6 87 18
```

Saída:

```
10, 31, 2
73, 24, 65
6, 87, 18
```

5. Se originalmente  $x = 4$ , qual é o valor de  $x$  após a avaliação de:
- a)  $x = 2$
  - b)  $x += 4$
  - c)  $x += x + 3$
  - d)  $x *= 2$
  - e)  $x /= x + 2$
6. Se originalmente  $x = 3$  e  $y = 4$ , quais são os valores de  $x$  e  $y$  após a avaliação de:
- a)  $x++ + y$
  - b)  $++x$
  - c)  $x++ + y++$
  - d)  $++x + 2$
  - e)  $x-- - y--$
7. Se originalmente  $x = 2$ ,  $y = 3$  e  $z = 1$ , qual é o valor de cada expressão a seguir:
- a)  $x + 2 / 6 + y$
  - b)  $y - 3 * z + 2$
  - c)  $z - (x + z) \% 2 + 4$
  - d)  $x - 2 * (3 + z) + y$
  - e)  $y++ + z-- + x++$

8. Escreva um programa que leia um *float* e depois extraia e imprima o dígito mais à direita da parte inteira dele.
9. Escreva um programa que leia um número em ponto flutuante representando um grau e depois converta e o imprima em radianos. Uma saída poderia ser

```
90 graus é1.57080 radianos
```

10. O quê será impresso quando o programa a seguir for executado usando 3 4 5 6 como dados?

```

#include <stdio.h>

int estranho(int x, int y){
    int t;
    int z;

    t = x + y;
    z = x * y;
    return (t + z);
} // estranho

int main(void){
    int a;
    int b;
    int c;
    int d;
    int r;
    int s;
    int t;
    int u;
    int v;

    scanf("%d %d %d %d", &a, &b, &c, &d);

    r = estranho(a, b);
    s = estranho(r, c);
    t = estranho(estranho(s, d), estranho(4, 2));
    u = estranho(t + 3, s + 2);
    v = estranho(estranho(estranho(u, a), b), c);

    printf("%d %d %d %d", r, s, t, u, v);
    return 0;
} // main

```

11. Escreva uma função para imprimir seu nome como mostrado a seguir. Escreva um programa completo para chamá-la a partir da função main.

```

*****

*                                                         *

*               Seu nome vai aqui                       *

*                                                         *

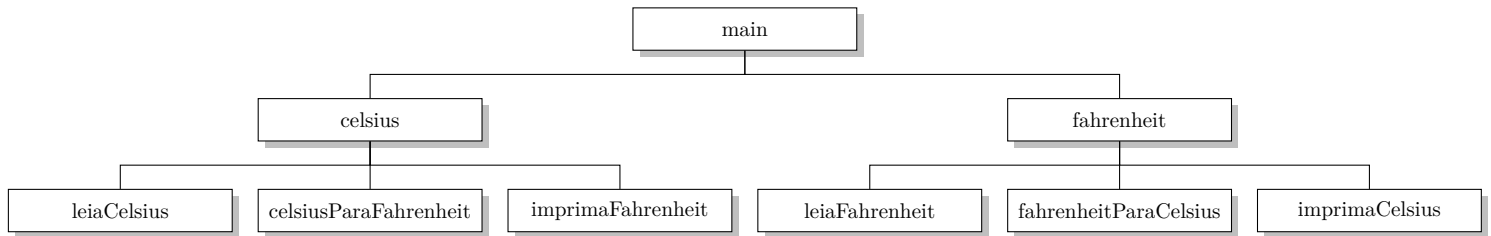
*****

```

12. Escreva uma função que converta polegadas em centímetros. Uma polegada é 2,54 centímetros. Depois escreva um programa que solicite do usuário uma medida em polegadas, chame a função de conversão e depois imprima o resultado em centímetros.
13. Escreva uma função que calcule o perímetro e a área de um triângulo retângulo dados os comprimentos da base e a altura.
14. A fórmula para converter graus Celsius em Fahrenheit é

$$F = 32 + C \frac{180.0}{100.0}$$

Escreva um programa que peça ao usuário para digitar a temperatura em graus Celsius e depois imprima o equivalente em graus Fahrenheit. Em seguida, ele pede ao usuário que digite um grau Fahrenheit e depois imprime o equivalente em graus Celsius. Execute o programa várias vezes. Teste também com valores negativos para as temperaturas. Use funções separadas em seu projeto, como mostrado a seguir:



15. Escreva uma função que encontre o menor entre quatro inteiros.
16. Transforme os laços **while** abaixo em laços **for** equivalentes.

a)

```

x = 0;
while (x < 10) {
    printf("%d\n", x);
    x++;
}
  
```

b)

```

scanf("%d", &x);
while (x != 9999) {
    printf("%d\n", x);
    scanf("%d", &x);
}
  
```

17. Transforme os laços **for** abaixo em laços **while** equivalentes.

a)

```

for(int x = 1; x < 100; x++)
    printf("%d\n", x);
  
```

b)

```

for( ; scanf("%d", &x) != EOF; )
    printf("%d\n", x);
  
```

18. O que será impresso por estes trechos de códigos?

a)

```

for(int x = 1; x <= 20; x++)
    printf("%d\n", x);
  
```

b)

```

for(int x = 1; x <= 20; x++){
    printf("%d\n", x);
    x++;
}
  
```

c)

```
for(int x = 20; x >= 10; x--)  
    printf("%d\n", x);
```

d)

```
for(int x = 20; x >= 1; x--){  
    printf("%d\n", x);  
    x--;  
}
```

e)

```
for(int x = 1; x <= 20; x++){  
    for(int y = 1; y <= 5; y++){  
        printf("%d\n", x);  
    }  
    printf("\n");  
}
```

f)

```
for(int x = 20; x >= 1; x--){  
    for(int y = x; y >= 1; y--){  
        printf("%3d\n", x);  
    }  
    printf("\n");  
}
```

19. Escreva um programa usando um laço **for** para imprimir uma linha com 60 asteriscos.
20. Escreva um programa usando um laço **for** que imprima os primeiros 50 números da sequência 1, 4, 7, 10, ....
21. O valor de  $\pi$  pode ser calculado pela seguinte fórmula

$$\pi = \sqrt{6 \left( \frac{1}{1^2} + \frac{1}{2^2} + \frac{1}{3^2} + \cdots + \frac{1}{\text{limite}^2} \right)}$$

Escreva uma função que leia o **double** `limite` e use essa fórmula para calcular pi. Depois escreva a função `main` para testar essa função, primeiro usando um limite de 5 termos e depois com 10 termos. Mostre o resultado de cada teste.

22. O que será impresso pelos programas a seguir?

a)

```
#include <stdio.h>  
  
int main(void){  
    int lista[10] = {0};  
  
    for(int i = 0; i < 5; i++){  
        lista[2*i+1] = i + 2;  
    }  
    for(int i = 0; i < 10; i++){  
        printf("%d\n", lista[i]);  
    }  
    return 0;  
}
```

b)

```

#include <stdio.h>

int main(void){
    int lista[10] = {2, 1, 2, 1, 1, 2, 3, 2, 1, 2};

    printf("%d\n", lista[2]);
    printf("%d\n", lista[ lista[2] ]);
    printf("%d\n", lista[ lista[2] + lista[3]]);
    printf("%d\n", lista[ lista[ lista[2]]]);
    return 0;
}

```

c)

```

#include <stdio.h>

int main(void){
    int lista[10] = {2, 1, 2, 4, 1, 2, 0, 2, 1, 2};
    int linha[10];

    for(int i = 0; i < 10; i++)
        linha[i] = lista[9 - i];
    for(int i = 0; i < 10; i++)
        printf("%d %d\n", lista[i], linha[i]);
    return 0;
}

```

23. Suponha que tenhamos dois arranjos A e B, cada um com 10 inteiros. Escreva uma função que teste se cada elemento do arranjo A é igual a seu correspondente no arranjo B. Em outras palavras, a função deve verificar se A[0] é igual a B[0], A[1] é igual a B[1] e assim em diante. A função deve retornar `true` se todos elementos são iguais ou `false`, em caso contrário.
24. Escreva uma função que reverte os elementos de um arranjo tal que o último elemento torne-se o primeiro, o penúltimo torne-se o segundo, etc. A função deve reverter os elementos **sem** usar um outro arranjo, mas você pode usar uma variável para manter um elemento temporariamente. Escreva a função principal para testar a sua função. Teste com um arranjo com um número par de elementos e também com um de número ímpar.
25. Faça um programa que leia um arranjo A contendo 10 números inteiros e depois determine e mostre quais elementos de A estão repetidos mais que duas vezes e quantas vezes cada um se repete.  
Por exemplo para a sequência de entrada 5, 4, 3, 18, 5, 3, 4, 18, 4, 18 a saída seria
  - 0 número 5 aparece 2 vezes.
  - 0 número 4 aparece 3 vezes.
  - 0 número 3 aparece 2 vezes.
  - 0 número 18 aparece 3 vezes.
26. Faça um programa que gere os dez primeiros números primos acima de 10000 e armazene-os em um vetor. Depois, imprima o arranjo resultante.
27. Faça um programa que leia um arranjo com 15 números inteiros. Depois da leitura, divida todos os elementos pelo maior valor presente no arranjo. Ao final, o programa deve mostrar o arranjo resultante.