

Ficha de Trabalho

Curso: Programação em C/C++ - fundamentos

UFCD/Módulo/Temática: UFCD0809_2N - Programação em C/C++

Ação: UFCD0809_2N

Formador/a: Hugo Dias

Data: 06/02/2025

Nome do Formando/a:

Funções/Procedimentos

1. No código abaixo a função *soma_valores* recebe dois valores inteiros e retorna a soma destes valores:

```
#include <stdio.h>

int soma_valores(int a, int b) {
    int c;
    c = a + b;

    return c;
}

int main() {
    int val1, val2, s;

    printf("Introduza valor:");
    scanf("%d", &val1);

    printf("Introduza outro valor:");
    scanf("%d", &val2);

    s = soma_valores(val1, val2);
    printf("Soma: %d", s);

    return 0;
}
```

Efetuar as seguintes alterações neste código:

- a) alterar a função *soma_valores* para que receba três valores inteiros e retorne a soma desses valores
- b) implementar a função *positivo* que recebe um valor inteiro e retorna o valor "1" no caso do valor ser positivo, caso contrário retorna o valor "0"

- c) implementar uma nova função: *maior_valor2*, que recebe dois valores inteiros e retorna o maior
- d) implementar uma nova função: *maior_valor3*, que recebe três valores inteiros e retorna o maior

2. Completar o código abaixo para que seja possível através da função *par_impar* verificar se o valor na variável *num* é par ou ímpar.

```
#include <stdio.h>

int main() {
    int num;

    printf("Introduza um valor:");
    scanf("%d", &num);
    .....

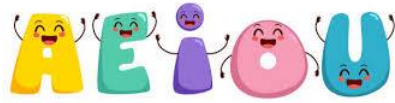
    return 0;
}

int par_impar(int x) {
    ???
}
```

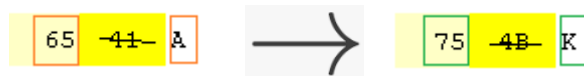
A função deve retornar "1" se o número é par.

- a) implementar uma nova função similar à função *par_impar* mas em vez de retornar um valor, escreve no terminal "Par" ou "Impar".

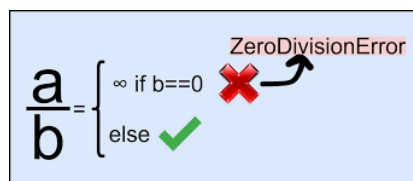
3. Implementar a função *vogal*, que retorna o valor "1" caso o caracter recebido seja uma vogal, caso contrário retorna o valor "0".



4. Desenvolver uma função que recebe um caracter e devolve o respetivo código *ascii*.
5. Implementar uma função que recebe o código *ascii* de um caracter e descodifica o caracter original de acordo com as seguinte regra:
- a. o código *ascii* recebido tem um incremento de dez unidades relativamente ao *ascii* original:



6. Desenvolver uma função, que recebe 2 valores e divide o primeiro pelo segundo, mas antes valida se o segundo valor é diferente de 0.



7. Implementar a função *Bissexto*, que recebe um valor que representa um ano, e retorna o valor *true* no caso do ano ser bissexto, caso contrário retorna *false*.

- Anos bissextos são aqueles que são múltiplos de 4, como 1996, 2000, 2004 (que podem ser divisíveis por 4)
- Porém, há uma exceção: múltiplos de 100 que não sejam múltiplos de 400

