



## UFCD 0783

### Programação em C/C++ - Funções e estruturas

#### Trabalho 2 - Funções sem passagem de parametros

1. Modifique o programa 6 do trabalho 1, de forma que as operações sejam realizadas cada uma por uma função para o efeito, exemplo para a **soma**:

Definição da função **soma()**

```
void soma()
{
    int v1,v2;
    printf("Insira v1: ");
    scanf("%d",v1);
    printf("Insira v2: ");
    scanf("%d",v2);
    printf("Resultado da soma %d\\",v1+v2);
}
```

Chamada da função **soma()** dentro do switch

```
case 1:
    soma();
    break;
```

2. Pretende-se construir um programa que faça uma contagem regressiva de 10 a 0, imprimindo os números da contagem com cores alternadas(vermelho e branco), no final do ciclo imprime com cor verde **Programa concluído com êxito!**

2.1 Construa o programa com base nos tópicos seguintes:

- Defina as cores ANSI

```
#define RESET    "\\033[0m"
#define RED     "\\033[31m"
#define WHITE   "\\033[37m"
#define GREEN   "\\033[32m"
```



- Crie uma função com o nome **countdown()** do tipo retorno **void**

Declare uma variável do tipo inteiro igual a 10, com o nome total

Estabeleça um ciclo **for** de total até 0

```
{  
    Use a instrução system("clear"), esta instrução irá limpar a consola (para usar esta  
    função inclua biblioteca stdlib.h)
```

Altere a cor verificando **se** a iteração é par ou ímpar

```
printf("%s%d%s\n", RED, i, RESET); // Número par: vermelho
```

A cada iteração use a instrução sleep(1); //Pausa de 1 segundo

```
}
```

Depois de terminar o ciclo imprima a mensagem **Programa concluído com êxito!** Com cor verde.

- Chame a função dentro do main

## 2.2 Analise o programa e responda às questões:

- De que forma a criação de funções ajuda a tornar o código mais modular e reutilizável?
- O que acontece caso a função countdown esteja depois da função main()
- Que bibliotecas são importadas no início do código?
- O que significa o tipo de retorno void na declaração da função countdown()?
- O que faz o comando system("clear") no programa?
- O que aconteceria se o comando sleep(1) fosse removido do programa? Qual seria o impacto na animação e na visualização das cores?
- O que aconteceria se a variável n fosse igual a 60?
- Analise o seguinte trecho de código e explique o que ele faz:

```
switch (i % 2)  
{  
    case 0:  
        printf("%s%d%s\n", RED, i, RESET);  
        break;  
    case 1:  
        printf("%s%d%s\n", WHITE, i, RESET);  
        break;  
}
```



### 3. Compile o seguinte programa

```
#include <stdio.h>
#include <unistd.h>
#include <stdlib.h>

// Definindo as cores ANSI
#define GREEN "\033[32m"
#define WR "\033[37;41m"

// Função para desenhar um quadrado Unicode
void desenharQuadrado() {
    printf("\u2588"); // Caractere Unicode para o quadrado preenchido
    fflush(stdout); // Garantir que o quadrado seja impresso imediatamente
}

// Função para animar barra de progresso
void barraDeProgresso() {
    int total = 120; // Define o número total de quadrados (tamanho da barra)
    for (int i = 1; i <= total; i++)
    {
        desenharQuadrado();
        // Alternar a velocidade
        if (i <= 30) {
            usleep(50000); //0,05 segundos a dormir
        }
        else if (i <= 60) {
            usleep(15000);
        }
        else if (i <= 72) {
            usleep(200000);
        }
        else if (i <= 74) {
            usleep(2000000);
        }
        else {
            usleep(5000);
        }
    }
    printf("\nDownload concluído!%s\n", GREEN);
    printf("%sESTE FICHEIRO PODE CONTER VÍRUS!!!!",WR);
}

int main() {
    barraDeProgresso(); // Chama a função para mostrar a barra de progresso
    return 0;
}
```

Modifique o programa:

- Mostrar um texto antes da animação, com uma pausa, e depois começar a barra.
- Imprimir um outro carater Unicode.
- Altere o tamanho da barra.
- Coloque mais um nível de velocidade.
- Modifique a cor do texto e do fundo dinamicamente durante a animação.
- Crie uma função chamada final\_msg() que mostre as mensagens finais do programa.