

TRABALHO 2

Exercício 1

```
#include <stdio.h>

void linha()
{
    for (int i=1;i<30;i++)
        printf("*");
}

void menu()
{
    linha();
    printf("\nMENU\n");
    linha();
    printf("\n1 - Soma");
    printf("\n2 - Subtração");
    printf("\n3 - Multiplicação");
    printf("\n4 - Divisão");
    printf("\n5 - Sair\n");
    linha();
}

void soma()
{
    int val1, val2;
    printf("\nInsira o primeiro valor: ");
    scanf("%d", &val1);
    printf("Insira o segundo valor: ");
    scanf("%d", &val2);
    printf("Resultado da soma: %d\n\n", val1+val2);
}

void sub()
{
    int val1, val2;
    printf("Insira o primeiro valor: ");
    scanf("%d", &val1);
    printf("Insira o segundo valor: ");
    scanf("%d", &val2);
    printf("Resultado da subtração: %d\n\n", val1-val2);
}

void mult()
{
    int val1, val2;
    printf("Insira o primeiro valor: ");
    scanf("%d", &val1);
    printf("Insira o segundo valor: ");
    scanf("%d", &val2);
    printf("Resultado da multiplicação: %d\n\n", val1*val2);
}

void divisao()
{
    int val1, val2;
    printf("Insira o primeiro valor: ");
    scanf("%d", &val1);
    printf("Insira o segundo valor: ");
    scanf("%d", &val2);
    if (val2 == 0) {
        printf("\nNão é possível efetuar a divisão por 0\n");
        printf("Insira um novo valor:\n ");
        scanf("%d", &val2);
        printf("Divisão = %d\n\n", val1/val2);
    } else {
        printf("Divisão = %d\n\n", val1/val2);
    }
}

int main()
{
    int op;

    do{
        menu();

        // printf("\n\nInsira o primeiro valor: ");
        // scanf("%d", &val1);
        // printf("Insira o segundo valor: ");
        // scanf("%d", &val2);

        printf("\nEscolha a opção: ");
        scanf("%i", &op);

        switch(op){
            case 1:
                soma();
                break;
            case 2: sub();
                break;
            case 3:
                mult();
                break;
            case 4: divisao();
                break;
        }
    } while (op != 5);

    return 0;
}
```

Exercício 2

```
Trab2exe2.c
1 #include <stdio.h>
2 #include <stdlib.h>
3 #define RESET "\033[0m"
4 #define RED "\033[31m"
5 #define WHITE "\033[37m"
6 #define GREEN "\033[32m"
7
8 void countdown()
9 {
10     int total=10,i;
11     for (i=total; i>=0 ; i--)
12     {
13
14         system("clear");
15
16         if (i%2==0 )
17             printf("%s%d%s\n", RED, i, RESET);
18         else
19             printf("%s%d%s\n", GREEN, i, RESET);
20         sleep(1);
21     }
22 }
23
24
25 int main()
26 {
27
28     countdown();
29     printf(GREEN "\nPrograma concluído com êxito", RESET);
30
31
32     return 0;
33 }
```

```
0
Programa concluído com êxito
...Program finished with exit code
Press ENTER to exit console.
```

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?**

Criar funções ajuda a organizar o código em pedaços menores com tarefas específicas. Isso deixa o código mais fácil de entender e de usar em outros lugares do programa, ou até em outros programas, sem ter que escrever tudo de novo.

No caso do `countdown()`, ele cuida só da contagem regressiva e da cor, separando essa lógica do resto do programa.

- **O que acontece caso a função `countdown` esteja depois da função `main()`**

Se a função estiver depois da `main()`, o compilador não reconhecerá a função no momento em que ela for iniciada no `main()`, gerando um erro de compilação.

- **Que bibliotecas são importadas no início do código?**

São as bibliotecas `#include` e `#include`

- **O que significa o tipo de retorno `void` na declaração da função `countdown()`?**

O `void` antes do `countdown()` quer dizer que essa função executa uma ação (no caso, mostrar os números e as cores). Porém, ela não devolve nenhum valor, apenas executa as ações.

- **O que faz o comando `system("clear")` no programa?**

O `system("clear")` limpa a tela, apagando o que estava escrito antes. Por isso, que a contagem regressiva aparece como se cada número surgisse numa nova tela, passando a ideia de uma animação.

- **O que aconteceria se o comando `sleep(1)` fosse removido do programa? Qual seria o impacto na animação e na visualização das cores?**

Sem o `sleep(1)`, os passos seriam executados de forma muito rápida, o que quebraria a animação e as cores não seriam percebidas pelo utilizador.

- **O que aconteceria se a variável `n` fosse igual a 60?**

A contagem regressiva seria mais longa, de 60 até 0.

O programa seria executado por 61 segundos, mostrando um número por segundo com as cores alternandas.

- **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;
}
```

O código verifica se (i) é par ou ímpar.

Caso seja par - mostra o número na tela em **vermelho**.

Caso seja ímpar - mostra o número na tela em **branco**.

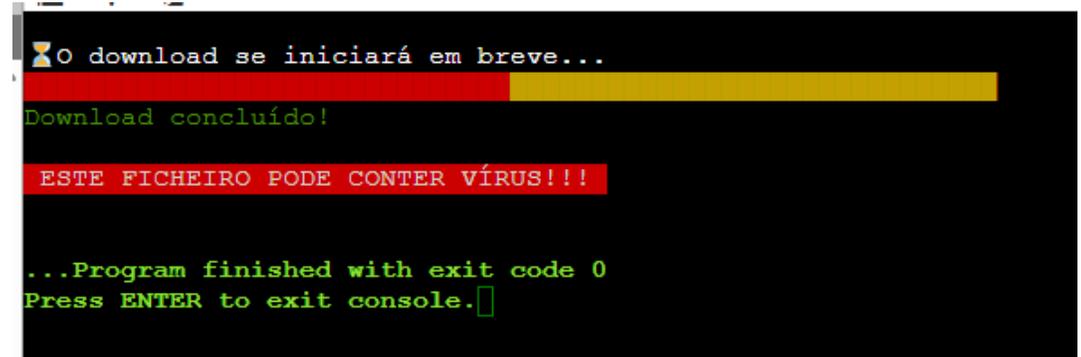
Já o `RESET`, é usado para interromper. Garantindo que a cor aplicada ao número (i) não continue afetando o texto seguinte na tela.

Exercício 3

```
1 #include <stdio.h>
2 #include <unistd.h>
3 #include <stdlib.h>
4
5 #define GREEN "\u033[32m"
6 #define RED "\u033[31m"
7 #define YELLOW "\u033[33m"
8 #define RESET "\u033[0m"
9 #define WR "\u033[37;41m"
10
11 void desenharQuadrado(int progress) {
12     if (progress <= 30) {
13         printf("%s\u2588%s", RED, RESET);
14     } else if (progress <= 60) {
15         printf("%s\u2588%s", YELLOW, RESET);
16     } else if (progress <= 80) {
17         printf("%s\u2588%s", GREEN, RESET);
18     }
19     fflush(stdout);
20 }
21
22 void final_msg() {
23     printf("%s \nDownload concluído!\n", GREEN, RESET);
24     printf("%s \nESTE FICHEIRO PODE CONTER VÍRUS!!!\n", WR, RESET);
25 }
26
27 void barraDeProgresso() {
28     //Mostrar um texto antes da animação, com uma pausa, e depois começar a barra.
29     //Imprimir um outro carater Unicode.
30     printf("\n\u23F3 O download se iniciará em breve...\n");
31     sleep(2);
32     int total = 60; // alterando o tamanho da barra
33
34     for (int i=1; i <=total; i++)
35     {
36         desenharQuadrado(i);
37         if (i <=30) {
38             usleep (50000);
39         }
40         else if (i <= 40) { //Novo nível de velocidade adicionado
41             usleep(10000);
42         }
43         else if (i <= 60) {
44             usleep(15000);
45         }
46     }
47 }
```

```
    }
    else if (i <=72) {
        usleep(200000);
    }
    else {
        usleep(5000);
    }
}
final_msg(); // Chama a função para mostrar as mensagens finais
}

int main() {
    barraDeProgresso();
    return 0;
}
```



```
⌚ O download se iniciará em breve...
Download concluído!
ESTE FICHEIRO PODE CONTER VÍRUS!!!
...Program finished with exit code 0
Press ENTER to exit console.
```