

Tema(s):

- Apontadores.
- Passagem de parâmetros.

Parte 1

1. Implemente o código presente nas páginas 11 e 12 dos slides teóricos. Efetue o *debug* e tente entender o funcionamento.
2. Complete o seguinte código seguindo as indicações dadas pelos comentários. Analise o comportamento do programa.

```
#include <stdio.h>

#include <stdio.h>

void foo(int a, int *aPtr) {

    // imprimir o valor de a
    // imprimir o endereço de a

    // imprimir o valor de aPtr
    // imprimir o endereço de aPtr

    // somar um ao valor de a
    // somar um ao valor apontado por de aPtr
}

int main() {
    int a = 0;
    // imprimir o valor de a
    // imprimir o endereço de a
    foo(a, &a);
    // imprimir o valor de a
    return 0;
}
```

3. Implemente um programa que leia 2 valores inteiros na função **main** e de seguida passe esses valores para a função **void somar(int num1, int num2, long *resultado)**. Essa função deve devolver a soma dos 2 valores no parâmetro **resultado**, que deverá ser apresentado na função **main**.
4. Implemente a função **void ordenar(int *v1, int *v2, int *v3)** que receba 3

apontadores para variáveis inteiras e ordene os valores pelas variáveis. Ex: **ordenar(1, 3, 2)** deverá atribuir a **v1** o valor **1**, **v2** o valor **2** e **v3** o valor **3**.

5. Implemente a função **int len(char *str)** que retorne o tamanho a *string* sem utilizar bibliotecas que não a **stdio.h**.

Parte 2

1. Implemente a função **int equal(char *str1, char *str2)** que retorne **1** se as strings forem iguais ou **0** se forem diferentes.
2. Implemente a função **void printFirst(char *str, int n)** que imprime caracter a caracter os primeiros **n** caracteres de uma *string*.
3. Implemente a função **char *strchar(char *str, char ch)** que retorne o endereço da primeira ocorrência de **ch** em **s**. Caso não exista, retorne **NULL**.
4. Implemente a função **int *findMax(int data[], int size)** que recebe um *array* e o tamanho deste. A função deve retornar o endereço de memória do elemento do *array* que tem o maior valor.
5. A função **swap_nums** parece funcionar, mas não **swap_pointers**. Corrija.

```
#include <stdio.h>

void swap(int *x, int *y) {
    int tmp = *x;
    *x = *y;
    *y = tmp;
}

void swap_pointers(char *x, char *y) {
    char *tmp;
    tmp = x;
    x = y;
    y = tmp;
}

int main() {
    int a = 0, b = 1;
    char *s1 = "da programação", *s2 = "Fundamentos";

    swap(&a, &b);
    printf("a: %d\n", a);
    printf("b: %d\n", b);

    swap_pointers(s1, s2);
    printf("%s %s\n", s1, s2);

    return 0;
}
```