

Programação 2023/24

LEI, LEI-PL, LEI-CE

Aula Laboratorial 3

Bibliografia:

K. N. King. *C programming: A Modern Approach* (2nd Edition). W. W. Norton: capítulos 11, 12 e 13.

Código de apoio para a aula:

<https://gist.github.com/FranciscoBPereira>

Ponteiros e Endereços Comunicação entre funções e manipulação de tabelas

Exercícios Obrigatórios

1. Execute a seguinte função para confirmar como os ponteiros em C armazenam endereços, garantindo assim uma forma alternativa para aceder a variáveis.

```
void f1(){
    int a=12, *p;
    float x=3.5, *q;

    p = &a;
    q = &x;

    printf("Valores: a=%d\tx=%.2f\n", a, x);
    printf("Valores: a=%d\tx=%.2f\n", *p, *q);

    printf("Endereco/Localizacao: a -> %p\tx -> %p\n", &a, &x);
    printf("Endereco/Localizacao: a -> %p\tx -> %p\n", p, q);
}
```

Programação 2023/24

LEI, LEI-PL, LEI-CE

2. Execute este código para confirmar como é efetuada a comunicação entre funções em C.

```
void f2(int x, int *p){
    printf("F2: Endereco/Localizacao: x=%p\tb=%p\n", &x, p);
    x++;
    (*p)++;
    printf("Valores em F2: a=%d\tb=%d\n", x, *p);
}

int main(){
    int a=1, b=2;

    printf("Valores Iniciais: a=%d\tb=%d\n", a, b);
    printf("Main: Endereco/Localizacao: a=%p\tb=%p\n", &a, &b);
    f2(a, &b);
    printf("Valores Finais: a=%d\tb=%d\n", a, b);
}
```

3. Escreva uma função em C que receba informação sobre 3 variáveis reais do tipo *float* e que efetue uma rotação entre elas, i.e., a segunda variável deve ficar com o valor da primeira, a terceira deve ficar com o valor da segunda e a primeira deve ficar com o valor da terceira. A função terá de alterar o valor das variáveis, pelo que os parâmetros da função são tipo ponteiro. O protótipo da função é o seguinte:

```
void rotacao(float *p1, float *p2, float *p3);
```

4. Escreva uma função em C que receba o nome e a dimensão de uma tabela de inteiros e verifique quantos dos seus elementos são pares e quantos são ímpares. Deve ainda calcular qual o maior valor presente na tabela e a posição onde ele se encontra.

Por exemplo, se o conteúdo da tabela for:

1	3	7	5	2	10	9	7	7	1
---	---	---	---	---	----	---	---	---	---

Existem 2 números pares, 8 números ímpares. O maior número é o 10 e está na posição 5.

A função recebe como argumentos um ponteiro para o início da tabela, o número de elementos que esta contém, um ponteiro para o inteiro onde deve colocar o n.º de elementos ímpares, um ponteiro para o inteiro onde deve colocar o n.º de elementos pares, um ponteiro para o inteiro onde deve colocar o maior valor armazenado na tabela e um ponteiro para o inteiro onde deve colocar a posição onde o maior valor se encontra. O seu protótipo é o seguinte:

```
void conta(int *t, int tam, int *np, int *ni, int *maior, int *pos);
```

Programação 2023/24

LEI, LEI-PL, LEI-CE

5. Escreva uma função em C que encontre os dois maiores elementos de um vetor de inteiros. A função recebe, como argumentos, o nome e a dimensão do vetor, e os ponteiros para as variáveis onde os dois maiores elementos devem ser armazenados. A tabela não está ordenada e pode assumir que não contém valores repetidos.

```
void procuraDupla(int *tab, int tam, int *prim, int *seg);
```

Exercícios Complementares

6. Escreva uma função em C que receba, como argumentos, o nome e a dimensão de uma tabela unidimensional de números inteiros e que coloque a zero todos os elementos cujo valor seja inferior à média dos valores armazenados nessa tabela. Pode assumir que quando a função for chamada a tabela já foi inicializada.

7. Escreva uma função em C que receba, como argumentos, os nomes e as dimensões de dois vetores de inteiros e verifique se estes são iguais. Considere que dois vetores de inteiros são iguais se tiverem o mesmo número de elementos e se, em posições equivalentes, tiverem elementos com o mesmo valor. A função devolve 1 se os vetores forem iguais, ou 0, no caso contrário.

8. Escreva uma função em C que receba 2 tabelas de inteiros e que contabilize quantos elementos elas têm em comum. A função tem o seguinte protótipo:

```
int comuns(int *tabA, int tamA, int *tabB, int tamB);
```

Recebe informação sobre as duas tabelas e devolve o número de elementos em comum. As tabelas recebidas estão ordenadas de forma crescente e dentro de cada uma das tabelas não existem elementos repetidos

9. Escreva uma função em C que receba um vetor de inteiros e mova todos os seus elementos algumas posições para a direita. O vetor deve ser considerado como sendo circular, ou seja, durante o deslocamento os últimos elementos voltam a entrar a partir do início. A função recebe, como parâmetros, o vetor, a sua dimensão e o número de posições a deslocar para a direita. Se o número de posições a deslocar for superior ou igual à dimensão do vetor, a função não efetua nenhuma alteração.

Por exemplo, se a função receber o vetor {2, 4, 6, 8, 10} com indicação para deslocar 2 posições: Deverá alterar o vetor para o seguinte conteúdo: {8, 10, 2, 4, 6}.

A função tem o seguinte protótipo:

```
void desloca(int *v, int tam, int shift);
```

Programação 2023/24

LEI, LEI-PL, LEI-CE

Trabalho Autónomo – Exercícios com Tabelas e Ponteiros

10. Qual o resultado da execução do seguinte programa?

```
#include <stdio.h>

int main() {
    int i, *p, *q;
    int a[5] = {0, 1, 2, 3, 4};

    for(p=a; p<= a+4; p++)
        printf("%d\t", *p);

    putchar('\n');

    for(p=a,i=1; i<3; i++)
        printf("%d\t", p[i]);
    putchar('\n');

    for(p=a, q=a+4; p<=q; p++, q--)
        printf("%d\t%d\t", *p, *q);
    return 0;
}
```

Programação 2023/24

LEI, LEI-PL, LEI-CE

11. Qual o resultado da execução do seguinte programa?

```
void f12A(char *a, char *b){
    do{
        b--;
        printf("%s\n", b);
    }while(a != b);
}

void f12B(char *a, char *b){
    while(a<b)
        printf("%s\n", a++);
}

int main(){
    char st1[10] = "123!";

    f12A(st1, st1+strlen(st1));
    f12B(st1, st1+strlen(st1));
    return 0;
}
```