

Programação 2024/25

LEI, LEI-PL, LEI-CE

Aula Laboratorial 4

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 Manipulação de Matrizes

Exercícios Obrigatórios

1. Escreva uma função em C que receba, como argumentos, o nome e as dimensões de uma matriz de números inteiros e mostre o seu conteúdo na consola. A função tem o seguinte protótipo:

```
void printMat(int nLin, int nCol, int m[][nCol]);
```

Repare na ordem pela qual surgem os argumentos da função. Só assim garante que a função consegue lidar corretamente com matrizes com dimensões diferentes.

2. Escreva uma função em C que calcule a média dos valores armazenados em cada uma das colunas de uma matriz de inteiros. A função recebe como argumentos o endereço inicial da matriz e as suas dimensões. Os valores calculados devem ser escritos na consola, com precisão de 2 casas decimais. A função deve devolver os índices das colunas com a média mais elevada e a média mais baixa.

3. Escreva uma função em C que efetue a transposição numa matriz de inteiros $N \times N$. A transposição consiste em trocar as linhas pelas colunas. Por exemplo:

Se tivermos a matriz				Após a transposição		
1	3	5	→	1	6	10
6	3	2		3	3	45
10	45	4		5	2	4

A função recebe como argumentos o endereço inicial da matriz e o valor N (pode assumir que a matriz é quadrada). A função tem o seguinte protótipo:

```
void tMat(int n, int mat[][n]);
```

Programação 2024/25

LEI, LEI-PL, LEI-CE

4. Escreva uma função em C que verifique se todos os elementos de uma matriz de inteiros são únicos. A função tem o seguinte protótipo:

```
int unicaMat(int nLin, int nCol, int mat[][nCol]);
```

A função devolve 1 se todos os elementos da matriz forem únicos, ou 0, caso contrário.

Exercícios Complementares

5. Pretende-se reduzir uma imagem a metade do seu tamanho original. A imagem inicial consiste numa tabela bidimensional com $M \times M$ valores inteiros onde cada *pixel* (ponto na imagem) pode tomar um valor inteiro entre 0 e 9 (tonalidades de cinza). A imagem reduzida é armazenada numa tabela bidimensional com $M/2 \times M/2$ valores reais onde cada elemento (*pixel*) corresponde à média dos 4 elementos que substitui. Para melhor compreensão do enunciado veja o exemplo apresentado de seguida (neste caso, para $M=8$):

Imagem original											
0	4	9	3	6	7	2	9				
9	4	7	8	3	4	7	2				
5	1	2	6	3	8	6	2				
7	1	4	8	7	2	0	7				
4	9	3	5	2	1	5	7				
5	3	4	7	8	9	1	0				
2	3	5	6	7	9	2	4				
2	3	4	5	6	7	8	9				
								Imagem reduzida			
								4.3	6.8	5.0	5.0
								3.5	5.0	5.0	3.8
								5.3	4.8	5.0	3.3
								2.5	5.0	7.3	5.8

Escreva uma função em C que efetue esta operação. A função recebe 3 argumentos: o valor M , o endereço inicial da matriz original e o endereço inicial da matriz reduzida.

6. Escreva uma função em C que receba uma matriz quadrada de inteiros e escreva os seus valores em formato espiral. A imagem seguinte ilustra a ordem pela qual os valores deverão ser escritos na consola:



A função tem o seguinte protótipo:

```
void espiral(int dim, int mat[ ][dim]);
```

Programação 2024/25

LEI, LEI-PL, LEI-CE

7. Considere que pretende resolver um quebra-cabeças que surgiu no seu jornal. Existe um retângulo, com um determinado número de linhas e de colunas, preenchido com caracteres alfabéticos em cada uma das suas posições. Na figura pode ver um exemplo para um quebra-cabeças com 5 linhas e 6 colunas.

e	B	a	u	l	q
l	e	r	r	s	s
u	w	u	q	g	r
a	a	l	l	u	a
p	m	h	u	d	j

Escreva uma função em C que procure todas as ocorrências de uma determinada palavra no quebra-cabeças. A palavra pode ocorrer numa linha ou numa coluna. De cada vez que a função encontrar uma ocorrência da palavra deve escrever no monitor o número da linha e da coluna em que a palavra tem início. Considerando o exemplo da figura, se a palavra a pesquisar for *lua* a função deveria escrever:

A palavra lua surge:

- Ao longo da coluna 0 com início na linha 1
- Ao longo da linha 3 com início na coluna 3

A função recebe como argumentos as dimensões e o endereço inicial da matriz de caracteres e um ponteiro para a palavra a pesquisar. O protótipo da função é o seguinte:

```
void procuraPal(int nLin, int nCol, char tab[][nCol], char *pal);
```

Ponteiros e Endereços Operações com Strings Constantes

Exercícios Obrigatórios

8. Escreva uma função em C que, para um dado mês recebido como parâmetro, escreva na consola a tradução para língua inglesa.

9. Escreva uma função em C que verifique se uma string recebida como parâmetro é o nome de um planeta do sistema solar. A função devolve 1 se a informação recebida corresponder a um planeta, ou 0, caso contrário.

Programação 2024/25

LEI, LEI-PL, LEI-CE

10. Um programa em C lida com uma estrutura de dados `char *s[][2]` para armazenar uma lista de sinónimos. A tabela é uma variável local da função `main()` e é inicializada na declaração. Os sinónimos armazenados são os seguintes:

estranho	bizarro
desconfiar	suspeitar
vermelho	encarnado
duvidar	desconfiar
carro	automóvel

- Escreva uma função em C que apresente na consola os pares de sinónimos armazenados na tabela. A função recebe como argumentos um ponteiro para o início da tabela e o número de pares de palavras aí armazenados (i.e., o número de linhas).
- Escreva uma função em C que verifique se uma determinada palavra tem um sinónimo conhecido. A função recebe, como argumentos, um ponteiro para o início da tabela, o número de pares de palavras armazenados e um ponteiro para a palavra a pesquisar. Devolve como resultado um ponteiro para o sinónimo da palavra recebida como argumento (se existirem vários sinónimos devolve um ponteiro para um deles). Se não existir nenhum sinónimo, a função devolve NULL.
- Escreva uma função em C que encontre a palavra alfabeticamente mais pequena que está armazenada na tabela. A função recebe, como argumentos, um ponteiro para o início da tabela e o número de pares de palavras armazenados. Devolve como resultado um ponteiro para a palavra que seja alfabeticamente menor.
- Escreva uma função em C que conte o número de ocorrências de cada vogal nas palavras armazenadas na tabela. A função recebe, como argumentos, um ponteiro para o início da tabela e o número de pares de palavras armazenados. Escreve na consola o número de ocorrências de cada vogal.
- Escreva uma função em C que verifique se existem palavras que apareçam em mais do que uma entrada na tabela. No exemplo em cima, a palavra `desconfiar` aparece duas vezes. A função recebe como argumentos um ponteiro para o início da tabela e o número de pares de palavras aí armazenados. Devolve como resultado o número de palavras que aparecem mais do que uma vez.

Programação 2024/25

LEI, LEI-PL, LEI-CE

- f) Escreva uma função em C que obtenha uma frase do utilizador e verifique se esta contém palavras que tenham sinónimo conhecido. Sempre que estes existirem, deve escrever no monitor a palavra original e o seu sinónimo. A função recebe como argumentos um ponteiro para o início da tabela e o número de pares de palavras aí armazenados.

Trabalho Autónomo – Exercícios com Matrizes

11. Qual o resultado da execução dos seguintes programas?

```
#include <stdio.h>

char* f1(char *a, char b){
    while((*a!='\0') && (*(a++) !=b))
        ;
    return a-1;
}

int main(){
    char *c[]={"abcdef", "ghijk", "lmnop"}, st[15] = "maio";
    int i;

    for(i=0; i<3; i++)
        puts(f1(c[i], st[i+1]));
    return 0;
}
```

```
#include <stdio.h>

int main(){
    int i, a[3][2] = {{1,2},{3,4},{5,6}};
    int *q, *p[3]={&a[0][0],a[1],a[2]+1};

    for(i=0; i<2; i++) {
        for (q = p[i]; q <= p[i+1]; q++)
            printf("%d\t", *q);
        putchar('\n');
    }
    return 0;
}
```

Programação 2024/25

LEI, LEI-PL, LEI-CE

Trabalho Autónomo – Parâmetros de um Programa (Argumentos da linha de comando)

Os parâmetros de um programa podem ser especificados no IDE *CLion* da seguinte forma:

Menu Run -> Edit Configurations -> Program Arguments

12. Escreva um programa em C que receba duas palavras a partir da linha de comando e escreva no monitor uma sequência constituída por caracteres retirados alternadamente de cada uma das palavras originais. As palavras especificadas devem ter o mesmo número de caracteres. Se isso não suceder (ou se o número de palavras for diferente de 2), o programa deve terminar imediatamente.

Exemplo: Se o programa receber AAA e BBB, a sequência final será: ABABAB.

13. Escreva um programa em C chamado *media* que receba na linha de comando um conjunto de inteiros e apresente a sua média na consola.

Por exemplo, se a chamada tiver o seguinte formato: `C:\> media 10 2 31 4 12`

Deve escrever o resultado: `11.8`

14. Escreva um programa em C que receba na linha de comando uma data no formato dd-mm-aaaa e a escreva na consola indicando o mês por extenso.

Por exemplo, se for indicada a data *12-01-2024* deve escrever *12 de janeiro de 2024*.