Aula prática #7 – Recursividade e Vetores

Problema 1

A sequência de Fibonacci é definida pela seguinte relação recursiva:

Função: $F_n = F_{n-1} + F_{n-2}$, com $F_0 = 0$ e $F_1 = 1$

Sem usar as estruturas de controlo de fluxo "for" e "while", implemente um algoritmo recursivo para calcular e imprimir a sequência de Fibonacci até um valor máximo ou até um número máximo de valores.
O resultado do programa deverá ser igual ao seguinte:

Exemplo

```
Pretende usar numero maximo de valores(1) ou valor maximo(2)? 1
Introduza um numero maximo de valores: 13
Sequencia: 0, 1, 1, 2, 3, 5, 8, 13, 21, 34, 55, 89, 144.

Pretende usar numero maximo de valores(1) ou valor maximo(2)? 2
Introduza o valor maximo: 20
Sequencia: 0, 1, 1, 2, 3, 5, 8, 13.
```

Problema 2

Implemente uma versão iterativa (usando "for"/"while") e compare o tempo de execução de ambos para valores de n superiores a 100000. Utilize a função $clock$ disponível na biblioteca $time$.

```
clock_t begin = clock();

/* chamada a funcao */

clock_t end = clock();
double time_spent = (double)(end - begin) / CLOCKS_PER_SEC;
```

Problema 3

Implemente uma função com base em variáveis estáticas que retorne o próximo número da sequência de Fibonacci, começando em F_2. Isto é, a primeira chamada à função deverá retornar 1, a segunda deverá retornar 2, a terceira deverá retornar 3, a quarta deverá retornar 5, etc. Implemente a função com o seguinte cabeçalho:

```
1  int proximoFib();
```

Problema 4

Implemente uma função recursiva que efetue a contagem do total de vogais presentes num vetor de caracteres. Para tal, utilize o seguinte protótipo, em que vec corresponde ao vetor e n ao número total de elementos de vec. Utilize o ficheiro `ex4.c`. Caso o tamanho do vetor seja inválido, a função deve retornar -1.

```
1  int totalVogais(char vec[], int n);
```

Exemplo

```
1  Vetor: { a d   3 B }
2  Total vogais: 1
```

Problema 5

Considere o ficheiro `ex5.c` e implemente a função `produto_vetores` com as seguintes caraterísticas:

- a função calcula o produto, elemento a elemento do vetor 1 pelo vetor 2
- os tamanhos dos vetores de entrada são indicados nos parâmetros n1 e n2
- o retorno da função deverá ser o tamanho do vetor de saída
- a função deve verificar se os tamanhos dos vetores de entrada estão dentro dos limites ($0 < n <= TAM\_MAX$; caso não esteja deve retornar -1
- caso os vetores de entrada tenham tamanhos diferentes, o vetor de saída deverá ter o tamanho do menor vetor de entrada

```
1  int produto_vetores(int entrada1[], int n1, int entrada2[], int n2, int saida[]);
```

Exemplo

```
vetor de entrada 1: { 4 7 6 }
vetor de entrada 2: { 2 8 2 1 3}
vetor de saida: { 8 56 12 }
```

Problema 6

Implemente uma função que procure e retorne todas as ocorrências de um dado elemento num vetor. Utilize para esse efeito a seguinte função:

```
int procuraTodos(int v[], int N, int x, int posicoes[]);
```

- v e N indicam o vetor e tamanho, respetivamente;
- x é o valor que se pretende procurar;
- posicoes é o vetor para guardar as posições das ocorrências encontradas;
- o valor de retorno deverá ser o número de elementos encontrados (dimensão do vetor posicoes) ou 0 se não encontrou nenhum elemento.

Implemente um programa para testar esta função.

Problema 7

Implemente uma função que remova elementos adjacentes iguais num vetor de caracteres. Utilize para esse efeito a seguinte função:

```
int compacta(char orig[], int N, char dst[]);
```

- orig e N indicam o vetor de entrada e tamanho, respetivamente;
- dst é o vetor onde será guardado o resultado;
- o valor de retorno deverá ser o tamanho do vetor dst.

Implemente um programa para testar esta função. Considere que a leitura do vetor termina com um ponto final ('.')

Input	Output
aaaaabbbbbcccc.	abc
abccccc.	abc
abc.	abc
carro.	caro

Problema 8

Escreva um programa que registe, num vetor multidimensional, o número mecanográfico e as notas obtidas por cada aluno. Cada um dos 4 alunos de uma turma realizou duas provas de avaliação. Imprima o conteúdo do vetor, bem como uma coluna extra onde mostre a média das notas obtidas por cada aluno.

Exemplo

```
Dados do aluno 1: 09001 5 15
Dados do aluno 2: 10701 10 12
Dados do aluno 3: 10321 16 17
Dados do aluno 4: 10452 8 11

Numero  Nota 1  Nota 2  Media
09001   5       15      10.0
10701   10      12      11.0
10321   16      17      16.5
10452   8       11      9.5
```

Problema 9

Tendo por base o programa do exercício 8, implemente uma função que ordene por ordem decrescente as notas obtidas pelos alunos. Utilize para esse efeito a seguinte função:

```
  void ordena(float notas[][2], int N);
```

As notas são guardadas numa matriz $N \times 2$, onde primeira coluna guarda o número do aluno e a segunda coluna a respetiva média. Altere o programa para testar esta função.

Exemplo

```
Numero Media
10321   16.5
10701   11.0
09001   10.0
10452   9.5
```

Problema 10

Implemente um programa que permita ao utilizador introduzir os valores de uma matriz quadrada de tamanho $N \times N$, com $N \leq 10$ (inicialmente especificado pelo utilizador). A matriz deverá ser guardada num vetor unidimensional e para conseguir fazer o mapeamento entre bi/unidimensional deverá usar a seguinte função:

```
1 int pos(int x, int y, int tamanho) { return x + y * tamanho; }
```

O programa deve permitir a realização das seguintes operações:

a) Visualizar a matriz. A visualização de uma matriz deve ser efetuada por uma função para esse efeito:

```
1 void imprimeMatriz(int matriz[], int N);
```

b) Multiplicar a matriz por um escalar (o valor em cada posição da matriz é multiplicado pelo escalar);

```
1 void produtoEscalar(int matriz[], int N, int escalar);
```

c) Multiplicação da matriz por uma matriz de $N \times 1$ (sendo N o tamanho do lado da matriz quadrada), cujos valores devem ser introduzidos pelo utilizador.

```
1 void multMatrizes(int matriz[], int mult[], int resultado[], int N);
```

Problema 11

O ficheiro `ex11.c` apresenta uma implementação parcial de um programa que testa algumas operações de processamento de imagem. A implementação fornecida lê um ficheiro com uma imagem e grava os resultados em imagens PNG. Para compilar, deverá usar o comando `clang ex7.c pnglib.c`; depois de executar o programa, a imagem original é gravada no ficheiro `resultado0.png`.

Considere que a imagem é representada por uma matriz com $altura \times largura$ pixéis. Tal como no problema anterior, essa matriz está guardada num vetor unidimensional, e portanto será necessário fazer o mapeamento entre linha/coluna e a respetiva posição no vetor. Nota: pode usar `POS(x,y)` para fazer esse mapeamento.

Implemente as seguintes funções, que transformam uma imagem de entrada numa de saída:

a) `filtro_media`: para cada pixel da imagem de saída, calcule o respetivo valor de acordo com:

$$S_{i,j} = \frac{E_{i-1,j-1}+E_{i,j-1}+E_{i+1,j-1}+E_{i-1,j}+E_{i,j}+E_{i+1,j}+E_{i-1,j+1}+E_{i,j+1}+E_{i+1,j+1}}{9}$$

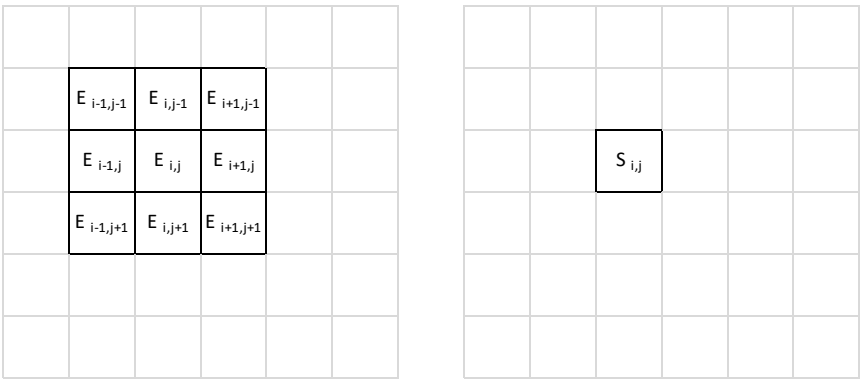

Imagem de entrada Imagem de saída

Nas linhas e colunas nos bordos da imagem de saída, onde não é possível usar todos os valores de entrada, não calcule os respetivos valores. A imagem de saída é guardada no ficheiro `resultado1.png` → qual é o efeito deste filtro?

b) `filtro_sobel`: para cada pixel da imagem de saída, calcule o respetivo valor de acordo com:

$$S_{i,j} = |-E_{i-1,j-1} + E_{i+1,j-1} - 2 * E_{i-1,j} + 2 * E_{i+1,j} - E_{i-1,j+1} + E_{i+1,j+1}|$$

Tal como na alínea anterior, não calcule os valores de saída nos bordos da imagem. A imagem de saída é guardada no ficheiro `resultado2.png` → qual é o efeito deste filtro?

c) `filtro_threshold`: para cada pixel da imagem de saída, calcule o respetivo valor de acordo com:

$$S_{i,j} = \begin{cases} 0, & E_{i,j} < threshold \\ 255, & E_{i,j} \geq threshold \end{cases}$$

A imagem de saída é guardada no ficheiro `resultado3.png` → qual é o efeito deste filtro?

Problema 12

O ficheiro `ex12.c` apresenta uma implementação parcial do jogo do galo ("Tic Tac Toe" em Inglês). Em primeiro lugar, analise em detalhe a lógica de jogo já implementada. De seguida, implemente as funções em falta no ficheiro de forma a garantir que o programa executa corretamente.

a) Implemente a função que imprime o tabuleiro de jogo:

```
1 void imprimeTabuleiro(char tabuleiro[LINHAS][COLUNAS])
```

b) Implemente a função que verifica se existe um vencedor. Esta deve retornar "1" em caso positivo ou "0" em caso negativo.

```
1 int haVencedor(char tabuleiro[LINHAS][COLUNAS])
```