

Ficha de Trabalho n.º 0 – 1.ª Ficha de Revisão

Objectivos: Revisão sobre ponteiros, memória dinâmica, usando exercícios de provas de AP.

1. Desenvolva uma função que dada uma matriz bidimensional $M \times N$ de inteiros indique se é esparsa. Se a matriz for esparsa é retornado 1 senão é retornado 0. Uma matriz é esparsa, se a percentagem de elementos nulos estiverem acima de um determinado limite (percentagem).

```
int Esparsa(int *Matriz, int M, int N, float percentagem)
```

2. Pretende-se analisar um conjunto de músicas. Cada música caracteriza-se pelo título, intérprete, ano, single (1:sim ou 0:não) e número de downloads nas plataformas aderentes.

Obs. 1. n , em todos os cabeçalhos das funções, representa o número de músicas no vetor lista;
2. utilize ponteiros em todas as alíneas para aceder ao valor dos atributos dos elementos do vetor.

- a) Defina o tipo de dados **Musica** e crie um vetor de nome lista, com um número de músicas definido pelo utilizador (valor disponível na variável nMusicas).

- b) Implemente uma função que escreva no ecrã, as características das músicas com mais do que um determinado número de downloads (parâmetro numD).

```
void musicasDownload(Musica *lista, int n, int numD)
```

- c) Desenvolva uma função que receba a listagem de músicas e devolva quantas têm um dado intérprete.

- d) Desenvolva uma função que guarde num ficheiro binário, as músicas singles, não singles ou ambas (parâmetro tpMeio='s', 'n' ou 'a', respetivamente), posteriores a um determinado ano. A função deve ainda devolver o número médio de downloads dessas músicas.

```
float gravaFichMusicas (Musica *lista, int n, int ano, char *nFich, char tpMeio)
```

Obs. Sintaxes: `FILE* fopen(const char* fileName, const char* mode);`
`size_t fwrite (const void* ptr, size_t size, size_t nElements, FILE* file);`

3. Pretende-se efetuar um estudo sobre os resultados de avaliação de um determinado número de alunos. Para cada aluno, regista-se o seu número mecanográfico, o seu nome e os resultados de avaliação de 5 UCs. Cada UC caracteriza-se pelo nome e respetiva avaliação.

Obs. 1. n , em todos os cabeçalhos das funções, representa o número de alunos no vetor lista;
2. utilize **ponteiros** em todas as alíneas para aceder ao valor dos atributos dos elementos do vetor.

- a) Defina o tipo de dados Aluno, UC e crie um vetor de nome ListAlun para guardar a informação necessária para este estudo, supondo que se pretendem guardar nReg elementos, sendo nReg especificado pelo utilizador.

- b) Para cada aluno, calcula-se a média das notas. Considera-se que um aluno transita se a média, arredondada à unidade, for igual ou superior a 10 valores. Defina a função qtosPassam que recebe a lista de alunos e calcula e devolve quantos alunos transitam.

- c) Determine o número médio de UCs, com notas inferiores a *lim* (e.g., 8) valores (considere esta média de UCs, como a soma total do número de UCs com classificação inferior a *lim*, a dividir pelo número de alunos).
- d) Elabore uma função que permita listar, num ficheiro de texto, os alunos com as UCs onde obtiveram (*val*) ou mais valores. No fich., deve haver uma linha com o nome do aluno e depois novas linhas para o nome e nota das UC que corresponderem à condição estabelecida. Depois, uma linha tipo "-----" como separador, seguindo-se um novo aluno e notas respetivas e assim sucessivamente.

```
void listaNotas(Aluno *lista, int n, int val, char *nomeFich)
```

Obs.: Em *nomeFich* está disponível o nome do ficheiro a usar.

Obs. Sintaxes: `FILE* fopen(const char* fileName, const char* mode);`
`fprintf (const void* ptr, ...);`

4. Pretende-se efetuar um estudo sobre a saúde de um determinado número de indivíduos, num dado ano. Para cada indivíduo, regista-se o seu número do Cartão de Cidadão, a sua altura, a sua massa, o género (M/F) e a data de nascimento (dia, mês, ano).

- Obs. 1. **n**, em todos os cabeçalhos das funções, representa o número de indivíduos no vetor *lista*;
2. utilize **ponteiros** em todas as alíneas para aceder ao valor dos atributos dos elementos do vetor.

- a) Defina o tipo de dados *Indiv*, *Data* e crie um vetor de nome *ListIndiv* para guardar a informação necessária para este estudo, supondo que se pretendem guardar *nReg* elementos, sendo *nReg* especificado pelo utilizador;
- b) Para cada indivíduo calcula-se o Índice de Massa Corporal (IMC) como $IMC = massa / altura^2$. Considera-se que um indivíduo é saudável, se o seu IMC estiver compreendido entre 18,5 e 24,9. Defina a função *indSaudaveis* que recebe a lista de indivíduos e calcula e devolve quantos indivíduos são saudáveis.
- c) Determine a média da massa corporal dos indivíduos nascidos entre dois anos especificados como parâmetros.
- d) Elabore uma função que permita remover do vetor um determinado indivíduo, devendo o vetor ficar ajustado, ou seja, não conter elementos vazios.

```
void removeIndiv(Indiv *vet_ind, int n, int nCC)
```

5. Crie um *main()* que permita testar todas as funções criadas.