Aulas T/TP – Exemplos Adicionais – Parte IV (aulas 7, 8 e 9)

Objetivos:

- Prepararação para o MiniTeste 2, especialmente na compreensão e utilização de:
  - Noção de Apontador (operadores & e *);
  - Aceder ao valor endereçado por um apontador;
  - Aritmética de apontadores;
  - Apontador para um vector.

Conceitos necessários à resolução da ficha: Diapositivos dos Caps.: ProgC-Apontadores&Estruturas.

Exercícios Propostos

1) Crie uma função (printStr) que escreva uma string, de um máximo 40 caracteres (um parâmetro da função), pela ordem de escrita e pela ordem inversa, utilizando aritmética de ponteiros.

```
// Função do exercício 2
void printStr(char *v1)
{
    int i=0;
    printf("\nFrase na ordem directa: ");
    do
    {
        printf("%c", *(v1+i));
        i++;
    } while (*(v1+i)!='\0');
    printf("\n");
    printf("\nFrase na ordem inversa: ");
    i--;
    do
    {
        printf("%c", *(v1+i));
        i--;
    } while (i>=0);
        printf("\n");
}
```

2) Crie uma função (strLen) que dada uma string (um parâmetro da função), calcule e mostre o respectivo tamanho. O tamanho deve ser obtido, utilizando ponteiros para a string.

```
// Função do exercício 3
int strLen(char *v1)
{
    int len=0;
    char *v2=NULL;
    v2 = v1;
    while (*v1!='\0') v1++;
    len=(int) (v1-v2);
    return len;
}
```

3) Crie uma função (strConcat) que dadas duas strings (parâmetros da função), junte as duas strings e disponibilize a string concatenada à função chamante na primeira string.

```
// Função do exercício 4
void strConcat(char *x, char *y) {
    while (*x!='\0')
        x++;
    while (*y!='\0')
        { *x=*y; x++; y++; }
    *x=*y;
}
```

4) Dadas as estruturas de dados abaixo apresentadas e MAX_V, elabore funções, usando ponteiros:

a) que permita ler n garrafas e registá-las no vector garrafeira, de tamanho MAX_V, sendo o nome da função leGarr. O valor de n deve ser introduzido pelo utilizador e deve estar no intervalo [1, MAX_V].

```
int leGrarr(GARRAFA *G, int nElem)
{
    int i;
    char marcG[50];
    for(i=0; i<nElem; i++)
    {
        printf("\n\nIntroduza os dados da garrafa %d...", i+1);
        printf("\nInsira o ano de produção: ");
        scanf(" %d", &G[i].ano); // ou scanf("%d", &(G+i)->ano);
        fflush(stdin);
        printf("\nInsira a Percentagem de Alcool: ");
        scanf(" %f", &G[i].def.percAlc); // ou scanf("%d",&(G+i)->def.percAlc);
        fflush(stdin);
        printf("\nInsira a cor do vinho (1-tinto; 2-branco, 3-rosé; 4-ruby): ");
        scanf(" %d", &G[i].def.cor); // ou scanf("%h",&(G+i)->def.cor);
        fflush(stdin);
        printf("\nInsira o tipo de vinho (M-mesa, G-generoso, E-Espumante, C-Champanhe): ");
        scanf(" %c", &G[i].def.tpVinho); // ou scanf("%c",(G+i)->def.tpVinho);
        fflush(stdin);
        printf("\nInsira a marca do vinho: ");
        gets(marcG);
        strcpy(G[i].def.marca, marcG);
        printf("\nInsira o preço da garrafa: ");
        fflush(stdin);
        scanf(" %f", &G[i].preco); // ou scanf("%f",&(G+i)->preco);
    }
    return i; // retorna o número de garrafas inserido
}
```

b) que calcule e disponibilize à função chamante os valores mínimo e máximo do preço das garrafas registadas no vector. O nome desta função deverá ser minMaxGarrafeira(...).

```
// função do exercício 6-b)
void minMaxGarrafeira(GARRAFA *F, float *maxVal, float *minVal, int nElem)
{
    int i;
    *maxVal=F->preco;
    *minVal=F->preco;
    for (i=0; i<nElem; i++)
    {
        if((F+i)->preco > *maxVal)
            *maxVal=(F+i)->preco;
        if((F+i)->preco < *minVal)
            *minVal=(F+i)->preco;
    }

}
```

c) que determine e devolva o preço e a marca da garrafa mais antiga. O nome desta função deverá ser precMarcaGVelha (...).

```
// função do exercício 6-c)
float precMarcaGVelha(GARRAFA *F, int nElem, char *nomeVel)
{
    int i=0, anoMin=0;
    float precVelho=0.0;
    anoMin=F->ano;
    precVelho=F->preco;
    strcpy(nomeVel, (F+i)->def.marca);
    for (i=0; i<nElem; i++)
    {
        if((F+i)->ano < anoMin)
        {
            anoMin=(F+i)->ano;
            precVelho=(F+i)->preco;
            strcpy(nomeVel, (F+i)->def.marca);
            printf("nome da garrafa mais antiga %s\n", nomeVel);
        }
```

```
    }
    return precVelho;
}

    #define MAX_V 20
    typedef struct caract{
        float percAlc; // % de alcool
        short cor; // 1-tinto; 2-branco, 3-rosé; 4-ruby;
        char tpVinho: // M-mesa, G-generoso, E-Espumante, C-Champanhe
        char nome[50];
    }CARACT;
    typedef struct garrafa
    {
        int ano;
        CARACT def;
        float preco;
    }GARRAFA;
```

5) Pretende-se armazenar a informação relativa a um conjunto de DVDs de Filmes:

a) Defina uma estrutura de dados capaz de armazenar a informação relativa a cada DVD - título do filme, actores principais, produtor, ano, preço e tipo (A-Acção; E-Espionagem; F-Ficção Científica; R-Romance; P-Policial; T-Terror; W-Western; O-Outro).

1.ª Solução para esta alínea

```
typedef struct dvd
{
       char titulo[30];
       char actores[100];
       char directores[60];
       short ano;
       float preco;
       char tipoF;
} DVD;

#define MaxPrat 10
#define MaxDVDPPrat 100
```

2.ª Solução para esta alínea

```
// actores principais, produtor, ano, preço e tipo (A-Acção; E-Espionagem; F-Ficção
Científica; R-Romance; PPolicial; T-Terror; W-Western; O-Outro).
#define MAX_TITULO_LEN 30
#define MAX_NOME_ATOR_LEN 30
#define MAX_ATORES_PRINCIPAIS_LEN 100
#define MAX_ATORES_FILMES 10

#define MAX_PRATELEIRAS 10
#define MAX_DVD_PRATELEIRA 100


typedef struct {
    char nome[MAX_NOME_ATOR_LEN];
    short age;
} ATOR;

#define MAX_TIPOS_FILME 8
typedef struct {
    char titulo[MAX_TITULO_LEN]; // título do filme,
    ATOR atores_principais[MAX_ATORES_FILMES]; // actores principais => se relevante
considerar estruturado e com caraterísticas próprias
    char produtor[MAX_TITULO_LEN];
    short ano;
    float preco;
    char tipo; // tipo (A-Acção; E-Espionagem; F-Ficção Científica; R-Romance; PPolicial;
T-Terror; W-Western; O-Outro).
} DVD;
```

b) Defina uma variável estruturada para armazenar a informação de um conjunto de DVDs, arquivados em 10 prateleiras com capacidade máxima de 100 DVDs, devendo definir os valores máximos de prateleiras e de DVDs por prateleira.

1.ª Solução para esta alínea

```
DVD dvds [MaxPrat][MaxDVDPPrat];
```

2.ª Solução para esta alínea (altera apena o nome da variável), mas é para ficar consistente com a 2.ª solução da alínea c)

```
DVD armario[MAX_PRATELEIRAS][MAX_DVD_PRATELEIRA];
```

c) Escreva uma função (usando ponteiros) de nome nFilmePTipo, que conte e devolva o número de filmes de cada tipo, entre anos, a especificar. Há um vector que indica quantos filmes existem em cada prateleira e outro com os tipos dos filmes.

1.ª Solução para esta alínea

```
short *nFilmePTipo(DVD *dvds, short anoI, short anoF, short *nFilmesPPrat, const char *tpFilme){
    short *nTotFPTipo;
    nTotFPTipo = (short *) calloc(8,sizeof(int)); // alocou-se memória para um vector de shorts
                                        // de 8 elementos para guardar o número total de cada tipo de DVD
    int i=0, j;

    for (i=0; i<MaxPrat; i++){
        for (j=0;j<*(nFilmesPPrat+i); j++)
        {
            if ((dvds+i*MaxDVDPPrat+j)->ano >= anoI && (dvds+i*MaxDVDPPrat+j)->ano <= anoF)
            {
                if ((dvds+i*MaxDVDPPrat+j)->tipoF == *tpFilme)
                    *nTotFPTipo=*nTotFPTipo+1;
                else if ((dvds+i*MaxDVDPPrat+j)->tipoF == *(tpFilme+1)){
                    *(nTotFPTipo+1)=*(nTotFPTipo+i)+1;}
                else if ((dvds+i*MaxDVDPPrat+j)->tipoF == *(tpFilme+2)){
                    *(nTotFPTipo+2)=*(nTotFPTipo+2)+1;}
                else if ((dvds+i*MaxDVDPPrat+j)->tipoF == *(tpFilme+3)){
                    *(nTotFPTipo+3)=*(nTotFPTipo+3)+1;}
                else if ((dvds+i*MaxDVDPPrat+j)->tipoF == *(tpFilme+4)){
                    *(nTotFPTipo+4)=*(nTotFPTipo+4)+1;}
                else if ((dvds+i*MaxDVDPPrat+j)->tipoF == *(tpFilme+5)){
                    *(nTotFPTipo+5)=*(nTotFPTipo+5)+1;}
                else if ((dvds+i*MaxDVDPPrat+j)->tipoF == *(tpFilme+6)){
                    *(nTotFPTipo+6)=*(nTotFPTipo+6)+1;}
                     else if ((dvds+i*MaxDVDPPrat+j)->tipoF == *(tpFilme+7)){
                         *(nTotFPTipo+7)=*(nTotFPTipo+7)+1;}
                else{
                    printf("\nTipo de filme %c não existente!! Considera-se tipo \"Outro\"\n",
                            ((dvds+i*MaxDVDPPrat+j)->tipoF == *(tpFilme+7)));{
                        *(nTotFPTipo+7)=*(nTotFPTipo+7)+1;}
                }
            }
        }
    }
    return nTotFPTipo;
}
```

2.ª Solução para esta alínea

```
// ANÁLISE
// Inputs: DVD dvds*, [const char* tipoFilmes, int numTipoFilmes], int ano_min, int ano_max
// Output: short* numFilmesPorTipo
// ['A', 'F', 'P']
// [10, 5, 3]

/**
 * @brief Determinar a posição em que o tipo de um dado DVD ocorre num array de tipos
 *
```

```
 * @param dvd DVD a considerar para análise
 * @param tipoFilmes Vetor 1D com os tipos de filmes a considerar
 * @param numTipoFilmes Número de elementos do vetor 1D tipoFilmes
 * @return int Posição em que o tipo do filme ocorre ou -1 caso não seja encontrado
 */
int determinar_posicao_tipo_filme(
    const DVD* dvd,
    const char* tipoFilmes,
    int numTipoFilmes
) {
    int i;
    char tipoFilme;
    for (i = 0; i < numTipoFilmes; i++) {
        tipoFilme = *(tipoFilmes + i);
        if (tipoFilme == dvd->tipo) {
            return i;
        }
    }
    return -1;
}

/**
 * @brief Efetuar a contagem do número de filmes por tipo de acordo com critérios especificados
pelo utilizador
 * (tipos de filmes a considerar, ano mínimo (inclusive), ano máximo (inclusive))
 * @param dvds Vetor 1D de DVDs (seguem a organização por prateleiras descrita atrás)
 * @param numDvdsPrateleiras Vetor 1D com o nº de DVDs contíguos por prateleira desde a primeira
posição (is. assume-se que não existem "espaços" entre DVDs em cada prateleira)
 * @param tipoFilmes Vetor 1D com os tipos de filmes que devem ser considerados para efeito das
contagens
 * @param numTipoFilmes Número de elementos do vetor 1D tipoFilmes
 * @param ano_min Ano mínimo (inclusive) de lançamento do filme
 * @param ano_max Ano máximo (inclusive) de lançamento do filme
 * @return short* Vetor com as contagens por tipo de filme pela ordem inferida do vetor
tipoFilmes e com o mesmo número de elementos
 */
short* nFilmePTipo(
    DVD* dvds,
    short* num_dvds_prateleiras,
    const char* tipo_filmes, int num_tipo_filmes,
    int ano_min, int ano_max
) {
    // Alocar espaço para as contagens de cada tipo de filme
    int i, p, d, posicao_tipo_filme;
    const DVD* dvd;
    short* numFilmesPorTipo;
    numFilmesPorTipo = (short*) malloc(num_tipo_filmes * sizeof(short));
    // Inicializar as contagens
    for (i = 0; i < num_tipo_filmes; i++) {
        *(numFilmesPorTipo + i) = 0;
    }
    // Percorrer cada prateleira
    for (p = 0; p < MAX_PRATELEIRAS; p++) {
        // Percorrer cada DVD da prateleira p
        for (d = 0; d < *(num_dvds_prateleiras + p); d++) {
            dvd = dvds + (p * MAX_PRATELEIRAS) + d;
            // Dado 1 array de tipos de filmes e um dvd => determinar a posição no array que
está o tipo do DVD
            posicao_tipo_filme = determinar_posicao_tipo_filme(dvd, tipo_filmes,
num_tipo_filmes);
            if (posicao_tipo_filme == -1) {
                continue; // Tipo do filme é irrelevante para contagem
            }
            if (dvd->ano < ano_min || dvd->ano > ano_max) {
                continue; // Filme fora do periodo pretendido
            }
```

```
            // Encontrei filme de tipo relevante => incrementar contagem
            *(numFilmesPorTipo + posicao_tipo_filme) += 1;
        }
    }
    return numFilmesPorTipo;
}
```

6) Crie o main() que lhe permita testar todas as funções criadas.

```
#include <stdio.h>
#include <stdlib.h>

int main(int argc, const char * argv[]) {

    // Exercício 1
    printf(("\nExercício 1\n"));
    char str1[40]={"A droga do dote e todo da gorda"};
    printStr(str1);

    // Exercício 2
    printf(("\nExercício 2\n"));
    char str2[40]={"A base do teto desaba"};
    printf("\nO tamanho da string %s é: %d\n", str2, strLen(str2));

    // Exercício 3
    printf(("\nExercício 3\n"));
    char s1[21],s2[11]; //21=10+10+1
    printf("\n Escreva uma frase (max. 10 caracteres): "); fflush(stdin);
    gets(s1);
    // fgets(s1, 20, stdin);
    printf("\n Escreva outra frase (max. 10 caracteres): "); fflush(stdin);
    // fgets(s2, 20, stdin);
    gets(s2);
    strConcat(s1, s2);
    printf("\n Concatenando, resulta: %s\n\n", s1);

    // Exercício 4
    // Alínea a)
    printf(("\nExercício 6-a)\n"));
    GARRAFA garrafeira[MAX_V];
    float maxVal=0.0, minVal=0.0;
    int nElem=3;


    //nElemRec=leGrarr(garrafeira, nElem);

    garrafeira[0].ano=2015;
    garrafeira[0].def.percAlc=13.4;
    garrafeira[0].def.cor=1;
    garrafeira[0].def.tpVinho='M';
    strcpy(garrafeira[0].def.marca, "Grão Vasco");
    garrafeira[0].preco=10.3;

    garrafeira[1].ano=2010;
    garrafeira[1].def.percAlc=14.2;
    garrafeira[1].def.cor=1;
    garrafeira[1].def.tpVinho='M';
    strcpy(garrafeira[1].def.marca, "Barca Velha");
    garrafeira[1].preco=400;

    garrafeira[2].ano=2018;
    garrafeira[2].def.percAlc=12.4;
    garrafeira[2].def.cor=2;
    garrafeira[2].def.tpVinho='M';
    strcpy(garrafeira[2].def.marca, "Aveleda-LOutreiro&Alvarinho");
    garrafeira[2].preco=5.3;



    printf(("\nExercício 4-b)\n"));
    minMaxGarrafeira(garrafeira, &maxVal, &minVal, nElem);
    printf("Valor da garrafa mais barata: %.2f; Valor da garrafa mais cara: %.2f\n", minVal, maxVal);

    printf(("\nExercício 4-c)\n"));
    char nomeVel[50];
    printf("Valor da garrafa mais antiga: %.2f; Marca da garrafa mais antiga: %s\n",
```

```
                                        precMarcaGVelha(garrafeira, nElem, nomeVel), nomeVel);

    // Exercício 5
    // Alínea b)
    printf(("\nExercício 5\n"));
    DVD dvds[MaxPrat][MaxDVDPPrat];
    short *nFilPPrat=NULL, *nTotFpTip=NULL;
    nFilPPrat = (short *) malloc (10 * sizeof(short)); // alocar o bloco de dados para o vector de shorts
    //short int nFilPPrat1 [10];                        // com o número de filmes que há por prateleira
    char *tpFilme;
    tpFilme = (char *) malloc (8 * sizeof(char));  // alocar o bloco de dados para conter o tipo de filmes
    strcpy(dvds[0][0].titulo, "E Tudo o Vento Levou");
    strcpy(dvds[0][0].actores, "Vivien Leigh, Clark Gable, Leslie Howard e Olivia de Havilland");
    strcpy(dvds[0][0].directores, "Victor Fleming, George Cukor e Sam Wood");
    dvds[0][0].ano=1939;
    dvds[0][0].preco=25.5;
    dvds[0][0].tipoF='R';
    //nFilPPrat1[0]=1;
    *nFilPPrat=1; // na primeira prateleira há um filme

    strcpy(dvds[1][0].titulo, "Casablanca");
    strcpy(dvds[1][0]. actores, "Humphrey Bogart e Ingrid Bergman");
    strcpy(dvds[1][0].directores, "Michael Curtiz");
    dvds[1][0].ano=1942;
    dvds[1][0].preco=30.5;
    dvds[1][0].tipoF='R';
    *(nFilPPrat+1)=1; // na segunda prateleira há um filme

    strcpy(dvds[2][0].titulo, "StarTrek");
    strcpy(dvds[2][0]. actores, "Chris Pine, Tom Cruise, Harrison Ford, Han Solo, Zachary Quinto");
    strcpy(dvds[2][0].directores, "J. J. Abrams");
    dvds[2][0].ano=2009;
    dvds[2][0].preco=25.0;
    dvds[2][0].tipoF='F';
    *(nFilPPrat+2)=1; // na terceira prateleira há um filme
    *(nFilPPrat+3)=0; // na quarta prateleira há 0 filmes
    *(nFilPPrat+4)=0; // na quinta prateleira há 0 filmes
    *(nFilPPrat+5)=0; // na sexta prateleira há 0 filmes
    *(nFilPPrat+6)=0; // na sétima prateleira há 0 filmes
    *(nFilPPrat+7)=0; // na oitava prateleira há 0 filmes
    *(nFilPPrat+8)=0; // na nona prateleira há 0 filmes
    *(nFilPPrat+9)=0; // na decima prateleira há 0 filmes

    //short nFilPPrat[8]={1, 1, 1, 0, 0, 0, 0, 0, 0, 0};
                    // há 1 filme na 1.ª, 2.ª e 3.ª prateleiras e 0 nas restantes.
    // *TpFilme (A-Acção; E-Espionagem; F-Ficção Científica; R-Romance;
    // P-Policial; T-Terror; W-Western; O-Outro).
    *tpFilme='A'; *(tpFilme+1)='E'; *(tpFilme+2)='F'; *(tpFilme+3)='R';
    *(tpFilme+4)='P'; *(tpFilme+5)='T'; *(tpFilme+6)='W'; *(tpFilme+7)='O';
    nTotFpTip=nFilmePTipo(&dvds[0][0], 1940, 2010, nFilPPrat, tpFilme);

    printf("\n\nEndereço do ponteiro no main   %p\n", nTotFpTip);

    int i=0;
    for (i=0; i<8; i++)
    {
        printf("(R) Número de filmes do tipo %c: %d\n", *(tpFilme+i), *(nTotFpTip +i));
    }
    printf("\n Programa terminado !!!\n");
    return 0;

}
```