

## Ficha de Trabalho N.º 4: Desafio

**Objetivo:** Permitir aos estudantes testarem a compreensão das bases da manipulação de strings, utilizando ponteiros para os parâmetros das funções, se tal for justificável.

### Preâmbulo

Para ajudar à resolução do problema colocado, vão-se apresentar 2 exercícios, cujas soluções se espera que permitam facilitar a descoberta da solução do exercício que constitui o desafio.

Exercício 1 (trata-se do exercício 2 da ficha 4):

Elabore um programa que leia uma string e escreva os dois primeiros caracteres no monitor, mostrando também o endereço da 1.ª posição da string. Use o operador [] e ponteiros para solucionar o problema.

### Solução

```
// Exercício 1)
// Elabore um programa que leia uma string e escreva os dois primeiros caracteres no monitor,
// mostrando também o endereço da 1.ª posição da string.
// Use o operador [] e ponteiros para solucionar o problema.
#include <stdio.h>
#include <stdlib.h>
#include <locale.h>
#define _CRT_SECURE_NO_WARNINGS
int main(void)
{
    setlocale(LC_ALL, "Portuguese");
    system("clear");
    char myString[20];
    printf("Escreva uma string: ");
    scanf("%s", myString);
    // solução isandso o operador []
    printf("\n Os dois primeiros caracteres sao %c %c \n", myString[0], myString[1]);
    printf("\n A Posição de memória do início de myString é: %p\n", &myString[0]);
    // solução usando ponteiros
    printf("\n Os dois primeiros caracteres sao %c %c \n", *myString, *(myString+1));
    printf("\n A Posição de memória do início de myString é: %p\n", myString);
}
```

## Exercício 2

Elabore um programa (não esqueça a respectiva função main()) que copie o conteúdo de uma string para outra, usando:

- uma função para o efeito, utilizando o operador [];
- uma função para o efeito, utilizando ponteiros.

```
// Exercício 2) Programa que copia o conteúdo de uma sting para outra, usando:
// a) uma função para o efeito, utilizando o operador [];
// b) uma função para o efeito, utilizando ponteiros.
#include <stdio.h>
#include <string.h>
#include <stdlib.h>
#include <locale.h>
#define _CRT_SECURE_NO_WARNINGS

void strCopiaOpParentesis(char x[], char y[]){
    int i=0;
    while (x[i]!='\0')
    {
        y[i]=x[i];
        i++;
    }
    y[i]=x[i];
}

void strCopiaComPonteiros(char *x, char *y){
    while (*x!='\0')
    {
        *y=*x; x++; y++;
    }
    *y=*x;
}

int main(void)// .....
{
    setlocale(LC_ALL,"Portuguese");
    system("clear");
    char c;
    char str1[21],str2[21];
    int i;
    printf("\nEscreva uma frase (max. 20 caracteres): ");
    fgets(str1, sizeof(str1), stdin);

    printf("\n*** Parte I - Resolução, usando o operador [].\n");
    printf("\nA string inserida foi: %s", str1);
    // solução, usando o operador []
    // mostra a string, caracter a caracter, separados por espaço
    i=0;
    printf("A string inserida (str1), mostrada caracter a caracter, separados por espaços
                                                foi:\n --> ");

    while(str1[i]!='\0'){
        printf("%c ", str1[i]);
        i++;
    }
    // invoca a função para efectuar a cópia
    strCopiaOpParentesis(str1, str2);
}
```

```
// mostra a string str2, depois da cópia
printf("A string str2, depois de efectuada a cópia é: %s\n", str2);

printf("*** Parte II - Resolução, usando ponteiros.....\n\n");
// solução usando ponteiros
// mostra a string str1, caracter a caracter, separados por espaço
char *x=str1;
printf("A string inserida (str1), mostrada caracter a caracter, separados por espaços

while(*x!='\0'){
    printf("%c ", *x);
    x++;
}
// invoca a função para efectuar a cópia
strCopiaComPonteiros(str1, str2);
// mostra a string str2, depois da cópia
printf("A string str2, depois de efectuada a cópia é: %s", str2);
// mostra a string str2, depois da cópia, caracter a caracter, separados por espaço
x=str2;
printf("A string str2, depois de copiada, mostrada caracter a caracter,

separados por espaços foi:\n --> ");

while(*x!='\0'){
    printf("%c ", *x);
    x++;
}
printf("\nDigite uma tecla para terminar...");
c=getchar();
return 0;
}
```

### Exercício 3 (desafio a solucionar)

a) Crie uma função `strModifyParImpar` que insira um caracter especificado nas posições par ou ímpares de um uma string.

Para ajudar, o cabeçalho da função apresenta-se infra:

```
/* Função que troca os elementos pares ou ímpares do vector str1 (string) pelo caracter c.
   Se ParOulmpar='p', troca o caracter existente nos elementos pares;
   */ se ParOulmpar='i', troca o caracter existente nos elementos ímpares.
```

```
void strModify(char *str1, char seParOulmpar, char c)
```

b) Copie a função anterior e altere-a de forma a trocar os elementos múltiplos de um número a especificar (parâmetro `m`) pelo caracter `c`, criando a função `strModifyM`.

c) A função anterior que se pretende mais genérica do que a criada na alínea a), poderá ser utilizada para substituir os elementos pares (especificando `m=2`), mas não os ímpares.

Copie a função anterior e altere-a de forma que já possa também substituir os números ímpares ou atendendo a um qualquer deslocamento (alterar o elemento de índice múltiplo de `m + d` (deslocamento)). P. ex., para substituir os números ímpares, bastaria especificar `m=2` e `d=1`. A nova função deverá chamar-se `strModifyMMaisD`.

O `main()` que permite testar as três funções criadas apresenta-se infra:

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

```
int main()
{
```

```
char vect[30]="Joao Francisco Silva";
strModify(vect, 'i', '+');
printf("Depois de executar a função strModify, a string é agora %s\n", vect);

strcpy(vect, "Joao Francisco Silva");
strModifyM(vect, 3, '+');
printf("Depois de executar a função strModifyM, a string é agora %s\n", vect);

strcpy(vect, "Joao Francisco Silva");
strModifyMMaisD(vect, 2, '+', 1);
printf("Depois de executar a função strModifyMMaisD, a string é agora %s\n", vect);
return 0;
}
```