INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA
SUL-RIO-GRANDENSE
Campus Charqueadas

Tecnólogo em Sistemas para Internet
Disciplina de Programação Estruturada – 2 Semestre
André Luís Del Mestre Martins

Lista de exercícios 2 – Ponteiros – **Prazo: combinado em aula**

INSTRUÇÕES:
A – Todos os arquivos compactados em formato ZIP (.zip). Inclua apenas os códigos-fonte (.c e .h), ou seja, não me envie os executáveis (.exe). O zip deve seguir o padrão: *[SEU_NOME].zip.*
B – Todas as funções solicitadas na lista de exercícios devem estar dentro de sua biblioteca. A biblioteca deve seguir o padrão: *[SEU_NOME].h* para os protótipos e *[SEU_NOME].c* para a descrição das funções.
C – Os códigos-fonte dos exercícios devem conter APENAS a função `main()` e devem seguir o padrão: *ex[NUMERO].c.* Lembre-se que são apenas 6 arquivos.
D – A lista de exercícios é entregue pelo Google Classroom.

PARTE 1 - PONTEIROS

1 – **Faca este exercício na função `main()`.** Escreva um programa para demonstrar a manipulação de ponteiros. Saída esperada:

```
 Endereco de NUM : 0x7ffcc3ad291c
 Valor de NUM : 29

 Agora PTR recebe o endereco de NUM.
 Endereco apontado pelo ponteiro PTR : 0x7ffcc3ad291c
 Conteudo do ponteiro PTR : 29

 O valor de NUM agora eh 34.
 Endereco apontado pelo ponteiro PTR : 0x7ffcc3ad291c
 Conteudo do ponteiro PTR : 34

 Agora o endereco apontado por PTR recebe 7.
 Endereco de NUM : 0x7ffcc3ad291c
 Valor de NUM : 7
```

2 – **Faca este exercício na função `main()`.** Escreva um programa para demonstrar o uso dos operadores &(endereco de) e *(valor no endereco). Saída esperada:

```
 Ponteiros : Demonstracao do uso dos operadores & e * :
-------------------------------------------------------

Inicializando variaveis
-------------------------------------------------------
 I = 300
 F = 300.600006
 C = z

 Usando operador & :
-----------------------
 Endereco de I = 0x7ffda2eeeec8
 Endereco de F = 0x7ffda2eeeecc
 Endereco de C = 0x7ffda2eeeec7

 Usando operadores & e * :
----------------------------
```

```
 Valor no endereco de I = 300
 Valor no endereco de F = 300.600006
 Valor no endereco de C = z

Inicializando ponteiros :
---------------------------------
 Endereco apontado por PTR_I = 0x7ffda2eeeec8
 Endereco apontado por PTR_F = 0x7ffda2eeeecc
 Endereco apontado por PTR_C = 0x7ffda2eeeec7

 Usando operador * :
---------------------------------
 Valor no endereco PTR_I = 300
 Valor no endereco PTR_F = 300.600006
 Valor no endereco PTR_C = z
```

3 – Escreva a função `void swap3v1(int*, int*, int*)` para trocar o conteúdo de três variáveis. Saída esperada:

```
 Antes de chamar funcao swap3v1 a=1, b=3 e c=5.
 Apos chamar funcao swap3v1 a=5, b=1 e c=3.
```

4 – Escreva a função `void swap3v2(int*, int*, int*)` para trocar o conteúdo de três variáveis usando OBRIGATORIAMENTE a função `void swap(int*, int*)` vista em aula. Saída esperada:

```
 Antes de chamar funcao swap3v2 a=1, b=3 e c=5.
 Apos chamar funcao swap3v2 a=5, b=1 e c=3.
```

PARTE 2 – PONTEIROS E VETORES

5 – Escreva a função `void copyArray(int* vetA, int* vetB, int tamanho)` para copiar um vetor para outro vetor utilizando ponteiros. Ambos vetores tem mesmo tamanho. Saída esperada:

```
 Vetor A = [1,2,3,4,5] Vetor B = [0,0,0,0,0].
 Copiando A em B...
 Vetor A = [1,2,3,4,5] Vetor B = [1,2,3,4,5].
```

6 – Escreva a função `void swapArray(int* vetA, int* vetB, int tamanho)` para trocar o conteúdo de um vetor por outro vetor utilizando ponteiros. Ambos vetores tem mesmo tamanho. Use OBRIGATORIAMENTE a função `void swap(int*, int*)` vista em aula. Saída esperada:

```
 Vetor A = [1,2,3,4,5] Vetor B = [0,0,0,0,0].
 Trocando A com B...
 Vetor A = [0,0,0,0,0] Vetor B = [1,2,3,4,5].
```

7 – Escreva a função `void reverseArray(int* vet, int tamanho)` para reverter o conteúdo de um vetor utilizando ponteiros.

```
 Vetor = [1,2,3,4,5]. Vetor reverso = [5,4,3,2,1].
```

8 – Escreva a função `int searchInArray(int* vet, int tamanho, int numero)` para procurar um número inteiro em um vetor de inteiros utilizando ponteiros. Retorne 1, caso o número seja encontrado. Caso contrário, retorne 0.

```
 Vetor = [1,2,3,4,5]. Procurar valor: 4.
 Saida: 1
 Vetor = [1,2,3,4,5]. Procurar valor: 0.
 Saida: 0
```