EXERCÍCIOS DE FIXAÇÃO Nº 02.1.1

Básico sobre ponteiros

1. Qual a função dos operadores **&** e **\*** quando associados a ponteiros? Exemplifique com código em C.

O operador “&” referencia especificamente à um endereço da memória atribuído à variável que o procede, enquanto o “\*” pode servir para inicializar um ponteiro, ou para se referir ao valor apontado dentro de um ponteiro.

Ex.: int n = 0;

int \*pN = &n; /\*(Ponteiro pN recebe o endereço de n.)\*/

printf(“%d”, \*pN); /\*(Exibe na tela o valor apontado por pN.)\*/

1. Por que é importante inicializar um ponteiro antes do seu uso?

É importante inicializar antes pois, caso contrário, o ponteiro não terá um endereço ao qual apontar e será nulo, gerando problemas na compilação do código.

1. As variáveis são sempre armazenadas nos mesmos endereços?

Não. Cada variável possui um endereço único na memória. Um ponteiro possui um endereço próprio e também um valor atribuído, que é o endereço de outra variável.

1. O que é indireção?

É o ato de referenciar algo indiretamente, como por exemplo no uso do operador “\*”, que ao ser usado junto a um ponteiro, irá referenciar o valor que está armazenado do endereço apontado pelo ponteiro, criando uma referência indireta. Ao atribuir um valor à um ponteiro utilizando este operador, o valor será salvo no endereço original salvo no ponteiro.

1. Como o compilador distingue o \* usado para a multiplicação do \* usado para “desreferenciamento” (acesso às informações existentes no endereço contido em um ponteiro) e do \* usado para declarar um ponteiro?

Através do uso de parêntesis. No caso de declarações, a explicitação do tipo de variável antecedida pelo operador acaba por fazer esta diferenciação. Ex.: int \*p; | \*p;

1. Escreva uma declaração de um ponteiro chamado **char\_ptr** para uma variável do tipo *char*.

char \*char\_ptr;

1. Se um programa contivesse uma variável *int* chamada **coast**, como você declararia e utilizaria um ponteiro chamado **p\_coast** para apontar para esta variável?

int \*p\_coast = &coast;

printf(“%d”, \*p\_coast);

1. Continuando com o exercício 13, como você atribuiria o valor 100 à variável **coast** usando acesso direto e indireto?

coast = 100;

(\*p\_coast) = 100;

1. Continuando com o exercício 14, como você imprimiria o valor do ponteiro juntamente com o valor para o qual ele está apontando?

printf(“Valor do ponteiro: %d.\n”, p\_coast);

printf(“Valor apontado pelo ponteiro: %d.\n”, \*p\_coast);

1. Mostre como atribuir o endereço de um valor do tipo *float*, chamado **radius**, a um ponteiro.

float \*pRadius = &radius;