**IF Sul-Rio-Grandense Campus Charqueadas**



**Curso Técnico de Nível Médio em Informática – Forma Integrada**

**Disciplina de Linguagem de Programação I**

**Lista de exercícios sobre ponteiros**

1. O que é e para que serve um ponteiro?

**Ponteiro é um tipo de váriavel que serve para armazenar o endereço de memória de uma outra váriavel cada uma do seu determinado tipo: int, char, float ou double.**

1. Qual a sintaxe correta para declarar um ponteiro?

**int \*p;**

**char \*p;**

**float \*p;**

**double \*p;**

1. Diferencie os operadores de ponteiros & e \* em C.

**&: Entendemos que o ponteiro acessa o endereço de memória da variável.**

**\*: É quando através do ponteiro acessamos o conteúdo de uma determinada variável, no qual este ponteiro aponta para está variável.**

1. Assumindo que o endereço da variável i foi atribuída ao ponteiro p, qual a forma correta de referenciar i através do ponteiro?

**\*p**

1. Explique qual a diferença entre um ponteiro do tipo float, int e char.

**A diferença é que ponteiro int só pode apontar para variáveis int, char somente para char, float somente para float e double somente para double.**

1. Supondo que c seja um ponteiro para char e i um ponteiro para int, explique a diferença entre realizar as operações i++ e c++

**C++ seria quando o ponteiro char avança uma posição a frente assim acessando o endereço de memória da próxima variável. EX: c=&vet[0] após o incremento fica c=&vet[1].**

**I++ seria quando o ponteiro int avança uma posição a frente assim acessando o endereço de memória da próxima variável. EX: i=&vet[0] após o incremento fica i=&vet[1].**

1. Supondo o código abaixo:

int \*p; int num; p = &num;

Diga se cada uma das comparações é verdadeira ou falsa, e justifique sua resposta:

* 1. num == &p;

**FALSO, pois não há valor definido para num, portanto não é possível afirmar que o valor de num é igual ao numero do endereço de memória que está armazenado o ponteiro p;**

* 1. num == \*p;

**VERDADEIRO, pois a variavel num tem o mesmo conteúdo de p, pois o ponteiro p tem como conteudo o valor de num.**

* 1. p == \*num;

**FALSO, pois no trecho de código o num não é um ponteiro**

* 1. p == &num;

**VERDADEIRO, pois no trecho de código o ponteiro está recebendo o endereço de memória de num, ou seja p é igual a num.**

1. O trecho de código abaixo falta uma linha para funcionar. Insira essa linha no código e justifique o porquê dela ser necessária:

int num, \*pont;

\*pont = 10;

printf("o número é %d", num);

**exe8\_listao\_ponteiros(resolução ta dentro da pasta)**

1. Escreva um trecho de código que divida o conteúdo de uma variável X por 2 sem utilizar o X na expressão.

**Exe9\_listao\_ponteiros(resolução ta dentro da pasta)**

1. Supondo que p seja um ponteiro para float, explique a diferença entre os comandos p++,

\*p++, (\*p)++, e \*(p++);

**p++:O ponteiro avança para a próxima posição do vetor, por exemplo: p=vet[0], após o incremento o ponteiro vai para o próxino endereço do vetor p=vet[1];**

**\*p++: Acessa o conteúdo da variável apontada pelo ponteiro e incrementa, por exemplo: se o conteúdo da variável é 12 após o incremento o conteúdo passa a ser 13.**

**(\*p)++: Mesma coisa que o anterior.**

**\*(p++): Avança para a próxima posição do vetor, e depois acessa o seguinte conteúdo.**

1. Explique qual a relação que um vetor tem com um ponteiro.

**A relação de um vetor com o ponteiro são as posições de memória, ou seja através de um ponteiro podemos percorrer um vetor acessando cada uma de suas posições.**

1. Supondo que vet seja um vetor, e p seja um ponteiro do mesmo tipo do vetor. Qual a diferença entre se fazer a atribuição p = vet e p = &vet[0]? Explique sua resposta.

**Nenhuma diferença, pois neste caso o ponteiro está recebendo a posição de memória inicial do vetor que seria a posição 0.**

1. Supondo que um ponteiro p esteja apontando para um vetor do tipo int. Logo após é apresentada a seguinte linha:

printf("%d",\*(p+7));

O que aparecerá na tela? Por quê?

**Irá aparecer o conteúdo da posição 8° posição do vetor**.

1. Supondo um vetor do tipo char chamado vet. A quinta posição de um vetor pode ser acessada pelo comando \*(vet+5)? Caso contrário justifique e sugira uma solução nos mesmos moldes. E se o vetor fosse do tipo int, o que mudaria? Justifique.

**Neste caso irá aparecer o conteúdo da sexta posição do vetor, ou seja se for para mostrar o conteúdo da posição 5, teria que ser \*(vet+4)**

1. Explique detalhadamente o que o trecho de código abaixo faz:

int vet[10],\*pv,i=0; pv = vet;

while (pv < &vet[10]){ i++;

\*pv = i; pv++;

}

Exe15\_listao\_ponteiros

1. O que a função sizeof() faz? O que irá retornar caso pedíssemos para mostrar na tela o valor de retorno de sizeof(int)? e de sizeof(char)? Explique.

**A função sizeof ela serve para mostrar o tamanho em bytes de um determinado tipo de variável, se retornarmos em sizeof(int) será 4 bytes, se for sizeof(char) será 1 byte!**

1. Usando a função sizeof(), como podemos saber qual o tamanho de um vetor qualquer, sabendo somente seu tipo e nome? Escreva a linha de código que retorna o tamanho deste vetor.

sizeof(vet)/sizeof(int)

1. O código abaixo contém um erro. Explique-o e faça a correção:

int \*p; int i; p = &i;

scanf("%d",p); //sem o \*, afinal p == &i

**Exe18\_listao\_ponteiros**

1. Como é feita a declaração de um ponteiro para ponteiro e para que eles servem? Dê um exemplo de uso.

|  |
| --- |
| ***int \*p; /// Declaração de um ponteiro***  ***int \*\*pp; /// Declaração de um ponteiro de ponteiro.***  ***int x=5;***  ***p=&x;/// O ponteiro recebe o endereço de memória da variável x, no entanto p tem o conteúdo de x.***  ***pp=&p;/// O ponteiro de ponteiro recebe o endereço de memória do ponteiro, no entanto ele pode acessar o conteúdo da variável.***  ***\*\*pp+=5;/// Acessando o conteúdo com \*\* acumulei 5 na soma somente para testar o acesso a variável atráves de um ponteiro ponteiro.***  ***printf("%i ",\*\*pp);/// Mostro no tela o conteúdo da variável x atráves do ponteiro de ponteiro.*** |

**Ponteiros de ponteiros apontam para um ponteiro, sendo assim acaba acessando o conteúdo de uma determinada variável podendo alterar o conteúdo desta maneira.**

1. Analise os dois códigos abaixo:

int x=10, \*p, \*q; p = &x;

q = p;

**No primeiro com a utilização de dois ponteiros eu consigo acessar o conteúdo da variável com qualquer um dos dois.**

int x=10, \*p, \*\*pp;

kk

N

**No segundo usando ponteiros de ponteiros podemos fazer da seguinte maneira.**

**p=&x;**

**pp=&p;**

**Agora com o ponteiro de ponteiro podemos manipular o conteúdo da variável x utilizando \*\***