

Ponteiros para variáveis

Conceito de Ponteiro

No C/C++ toda variável ao ser declarada possui também um endereço de memória que pode ser acessado através do **caractere &**.

- Considere a declaração: **int aux = 10;** => O nome do endereço de memória de **aux** será definido automaticamente como **&aux**
- Para exibir o conteúdo de **aux** => **cout << aux ;**
- Para exibir o endereço de memória de **aux** = **cout << &aux ;**

Declaração de um Ponteiro

Ponteiro é uma variável que serve para armazenar o endereço de outra variável, exatamente do mesmo tipo. Todo ponteiro, ao ser criado obrigatório deve ser inicializado antes de ser usado.

- Criando o ponteiro *ptx

int *ptx;

- Inicializando o ponteiro *ptx, isto é, apontando ptx para um endereço de memória

ptx = &aux; ou ptx = NULL;

Programa 1

```
#include "iostream"
#include "cstdlib"
using namespace std;
int main () {  setlocale(LC_ALL, "Portuguese");
int aux = 10; // cria uma variável inteira
int *ptaux; // cria um ponteiro inteiro
ptaux = &aux; // inicializa o ponteiro
cout<< "\nO valor de aux é:"<< aux;
cout<< "\nO endereço de memória de aux é :"<< &aux;
cout<< "\nO endereço de memória de aux é :"<< ptaux;
cout<< "\nO endereço de memória ptaux é:" << &ptaux;
cout<< "\nO valor de aux é:" << *ptaux;
system("pause"); return 0; }
```

Todo ponteiro deverá ser inicializado logo depois de criado, um ponteiro têm que ser exatamente do tipo da variável que ele irá armazenar o valor ou apontar.

No programa ao lado, a variável **aux** têm um endereço de memória chamado **&aux** e um ponteiro apontando para ela que é o ***ptaux**.

Para apontar um ponteiro para o endereço de uma variável, o asterisco deve ser retirado.

ptaux = &aux

Exemplo Interativo de Ponteiros

Crie um Algoritmo que armazene dois valores e faça a troca de valores, entretanto, utilize ponteiros como parâmetros das sub rotinas envolvidas.

Programa Interativo6

```
#include "iostream"
#include "cstdlib"
using namespace std;

void trocar (int *x, int *y) { int aux;
    aux = *x;
    *x = *y;
    *y = aux; }

int main() {

    int a = 10,      b = 20;
    cout << "\nValor inicial de A:" << a ;
    cout << "\nValor inicial de B:" << b ; system("pause");
    trocar ( &a, &b );
    cout << "\nValor Final de A:" << a ;
    cout << "\nValor Final de B:" << b ; system("pause");
}
```

O void **trocar** () possui dois ponteiros como argumentos, isso significa que teremos que enviar dois endereços de memória um para o ponteiro ***x** e outro para o ponteiro ***y**.

A variável **aux** recebe o conteúdo do **ponteiro *x**. o **ponteiro *x** recebe o conteúdo do **ponteiro *y**, o **ponteiro *y** recebe o conteúdo de **aux** que é um número qualquer.

No **main()**, ao executar o **void trocar ()** os endereços de **&a** e **&b** são enviados para dentro dos argumentos ***x** e ***y**.

TAREFA / AVALIAÇÃO CONTINUADA

I - Responda as questões a seguir

1. Defina o que seria um ponteiro. Explique e dê exemplos.
2. O que seria o endereço de uma variável. Explique como é criado e como posso exibi-lo.
3. Como posso acessar o conteúdo de uma variável usando um ponteiro. Explique e dê exemplo.
4. Como posso acessar o endereço de uma variável usando ponteiro. Explique e dê exemplo.
5. Qual seria do tipo de ponteiro para a STRUCT PESSOA?
6. Declare duas variáveis e dois ponteiros. Exiba o resultado da soma dessas variáveis somando os seus ponteiros (com Asterisco).

II - Escolha dois dos três exercícios para utilizar funções que tenham argumentos como ponteiros. Faça os programas com estrutura case, menu , funções de entrada, de cálculos e de saída. a) Equação do Segundo Grau b) Triângulo Heron c) Cálculo da área de um círculo.

NOTA: Entregar o código fonte e o print de execução dentro de um único arquivo DOCX ou PDF, somente através do Teams.