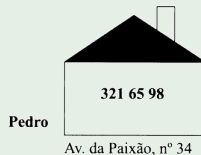


Analogia das Casas

Imagine um universo entre o passado e o futuro onde cada casa possui por direito um telefone fixo(esqueça celulares).

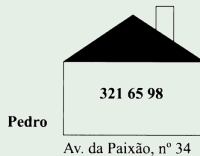
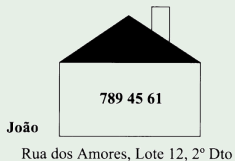
Exemplo



Analogia das Casas

Telefone Joao , Ana , Pedro ;

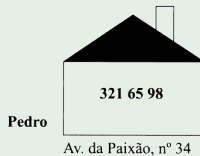
Exemplo



Analogia das Casas

```
Printf("%d", Joao)  
Printf("%d", Ana)  
Printf("%d", Pedro)
```

Exemplo



Analogia das Casas

<code>Printf("%d", Joao)</code>	→	789 45 61
<code>Printf("%d", Ana)</code>	→	456 78 12
<code>Printf("%d", Pedro)</code>	→	321 65 98

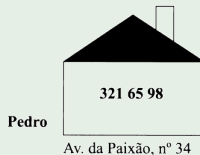
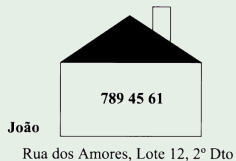
Exemplo



Analogia das Casas

Em C também temos o operador `&` que retorna um endereço de uma variável.

Exemplo



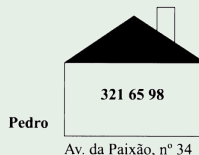
Analogia das Casas

```
Printf("%s", &Joao)
```

```
Printf("%s", &Ana)
```

```
Printf("%s", &Pedro)
```

Exemplo



Analogia das Casas

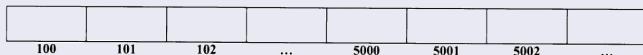
`Printf("%s", &Joao)` → Rua dos Amores, Lote 12, 2Dto.
`Printf("%s", &Ana)` → Calçada da Amizade, n 23, 7 B.
`Printf("%s", &Pedro)` → Av. da Paixao, n 34.

Exemplo



Memoria RAM

- A Memoria RAM pode ser vista como um enorme vetor de bytes consecutivos.



Memoria RAM

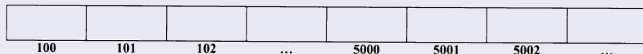
- A Memoria RAM pode ser vista como um enorme vetor de bytes consecutivos.



- Cada um desses bytes ocupa uma posição identificada por um numero.

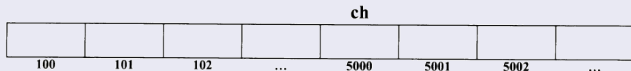
Memoria RAM

- A Memória RAM pode ser vista como um enorme vetor de bytes consecutivos.



- Cada um desses bytes ocupa uma posição identificada por um numero.
- ao declararmos uma variável, o processador usa seu tipo e nome para escolher uma posição na RAM.

```
char ch;
```



Ponteiro

- Um ponteiro é um tipo de dado especial que guarda endereços (a posição na RAM basicamente).

Ponteiro

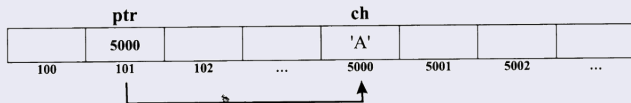
- Um ponteiro é um tipo de dado especial que guarda endereços (a posição na RAM basicamente).
- Suponha que **ptr** é um ponteiro e fazemos as seguintes ações:

```
ch = 'A';  
ptr = &ch;
```

Ponteiro

- Um ponteiro é um tipo de dado especial que guarda endereços (a posição na RAM basicamente).
- Suponha que **ptr** é um ponteiro e fazemos as seguintes ações:

```
ch = 'A';  
ptr = &ch;
```



Ponteiro

- Um ponteiro ainda é uma variável como outra qualquer.

Ponteiro

- Um ponteiro ainda é uma variável como outra qualquer.
- A sintaxe é a seguinte:

```
tipo * nome
```

Ponteiro

- Um ponteiro ainda é uma variável como outra qualquer.
- A sintaxe é a seguinte:

`tipo * nome`

`nome` : Nome dado a variável.

`tipo` : Tipo de dados da variável que está apontará.

`*` : Indica que esta variável é um ponteiro.

Ponteiro

- Um ponteiro ainda é uma variável como outra qualquer.
- A sintaxe é a seguinte:

`tipo * nome`

nome : Nome dado a variável.

tipo : Tipo de dados da variável que está apontará.

***** : Indica que esta variável é um ponteiro.

Exemplos

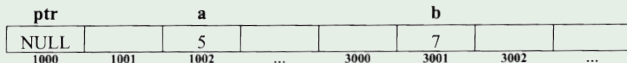
```
char a, b, c, *p, *t;  
int *pi, i, *pj, j;
```

Exemplo

```
int a = 5, b = 7;  
int * ptr = NULL; //NULL e o endereço nulo
```

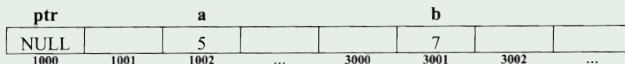
Exemplo

```
int a = 5, b = 7;  
int * ptr = NULL; //NULL e o endereço nulo
```



Exemplo

```
int a = 5, b = 7;  
int * ptr = NULL; //NULL e o endereço nulo
```



```
ptr = &a;
```

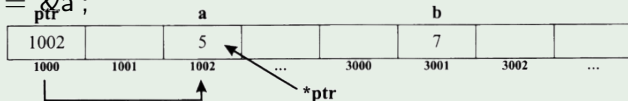
Exemplo

```
int a = 5, b = 7;
```

```
int * ptr = NULL; //NULL e o endereço nulo
```

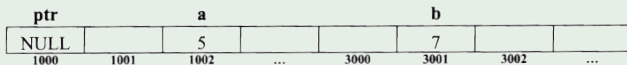


```
ptr = &a;
```

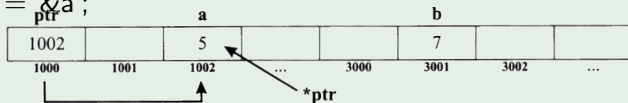


Exemplo

```
int a = 5, b = 7;  
int * ptr = NULL; //NULL e o endereço nulo
```



```
ptr = &a;
```



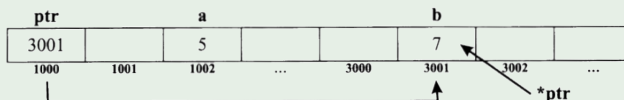
```
print("%d-%d-%d", a, b, *ptr); //5 7 5
```

Exemplo

```
ptr = &b;
```

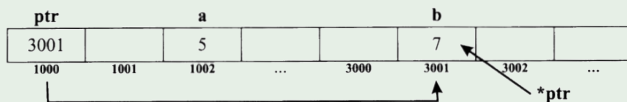
Exemplo

```
ptr = &b;
```



Exemplo

```
ptr = &b;
```

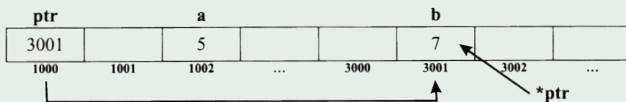


```
print("%d-%d-%d", a, b, *ptr);    //5 7 7
```

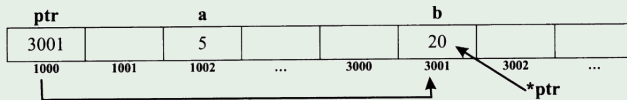
```
*ptr = 20;
```

Exemplo

```
ptr = &b;
```

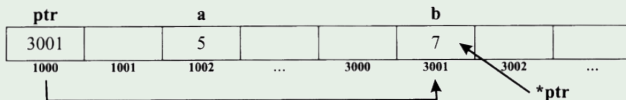


```
print("%d-%d-%d", a, b, *ptr);    //5 7 7  
*ptr = 20;
```



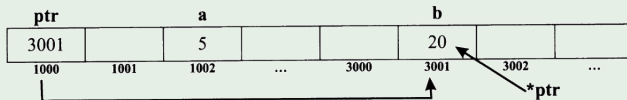
Exemplo

```
ptr = &b;
```



```
print("%d-%d-%d", a, b, *ptr);    //5 7 7
```

```
*ptr = 20;
```



```
print("%d-%d-%d", a, b, *ptr);    //5 20 20
```

Tamanhos

- "Por que ponteiros tem tipo?".

Tamanhos

- "Por que ponteiros tem tipo?".
- Nos exemplos anteriores vimos cada variável ocupando um espaço na memória, mas na pratica, não é bem assim.

Tipos de dados

Tamanhos

- "Por que ponteiros tem tipo?".

Exemplo

```
char a = 'Z';  
int n = 1234;  
float pi = 3.1415;  
  
char *ptr_a = &a;  
int *ptr_n = &n;  
float *ptr_pi = &pi;
```

Tipos de dados

Tamanhos

- "Por que ponteiros tem tipo?"

Exemplo

```
char a = 'Z';
```

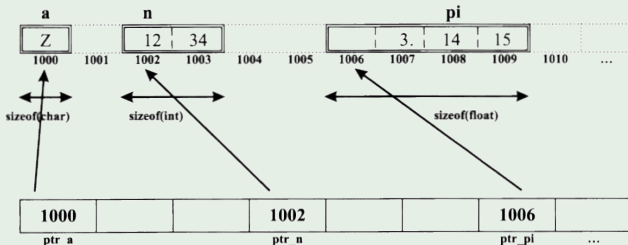
```
int n = 1234;
```

```
float pi = 3.1415;
```

```
char *ptr_a = &a;
```

```
int *ptr_n = &n;
```

```
float *ptr_pi = &pi;
```



Operações

Operação	Exemplo
Atribuição	<code>ptr = &x</code>
Incremento	<code>ptr = ptr+2</code>
Decremento	<code>ptr = ptr-3</code>
Apontado por	<code>*ptr</code>
Endereço de	<code>&ptr</code>
Diferença	<code>ptr1 - ptr2</code>
Comparação	<code>ptr1 > ptr2</code>

codigo

```
char s[20] = "frango-frito";  
char *pt = s;
```

Relações Importantes

- $s == \&s[0]$
- $s[1] == *(s+1)$

codigo

```
char s[20] = "frango-frito";  
char *pt = s;
```

Relações Importantes

- $s == \&s[0]$
- $s[1] == *(s+1)$

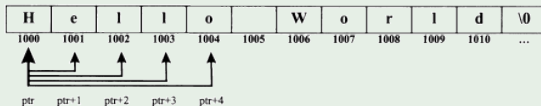
Formas de acessar o caractere 'a'

$s[2]$	Acessa a posição de índice 2
$*(ptr+2)$	conteúdo apontado por $ptr+2$
$*(s+2)$	conteúdo apontado por $ptr+2$
$ptr[2]$	podemos usar ponteiros de forma semelhante a vetores

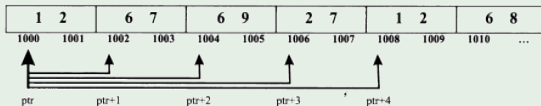
Vetores e Ponteiros

Exemplo

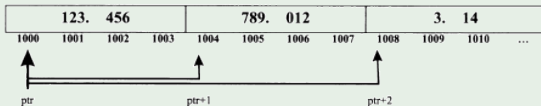
Se *ptr* for um ponteiro para um vetor de caracteres, então



Se *ptr* for um ponteiro para um vetor de inteiros (dois *bytes* cada), então



Se *ptr* for um ponteiro para um vetor de *floats* (quatro *bytes* cada), então



Pontos Importantes

- Como já foi visto nos exemplos anteriores, podemos utilizar os ponteiros como parâmetros de funções.

Pontos Importantes

- Como já foi visto nos exemplos anteriores, podemos utilizar os ponteiros como parâmetros de funções.
- Com isso podemos acessar um vetor em uma função, utilizando somente o endereço dele.

Pontos Importantes

- Como já foi visto nos exemplos anteriores, podemos utilizar os ponteiros como parâmetros de funções.
- Com isso podemos acessar um vetor em uma função, utilizando somente o endereço dele.
- Isso também nos dá a habilidade de mudar variáveis que não foram criadas em uma função.

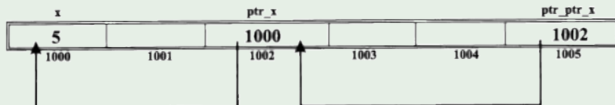
Exemplo

```
int x = 5;  
int * ptr_x = &x;  
int ** ptr_ptr_x = &ptr_x;
```

Ponteiros de Ponteiros

Exemplo

```
int x = 5;  
int * ptr_x = &x;  
int ** ptr_ptr_x = &ptr_x;
```



Exemplo

```
int x = 5;  
int * ptr_x = &x;  
int ** ptr_ptr_x = &ptr_x;
```

Expressão	tipo	valor
x	int	5
ptr _x	int *	1000
*ptr _x	int	5
ptr _{ptr_x}	int **	1002
ptr _{ptr_x}	int	1000
**ptr _{ptr_x}	int	5