



A LANGUAGE FOR AVR

Parte 3

© Prof. Eng^o esp Luiz Antonio Vargas Pinto
www.vargasp.com

Vetores e Matrizes

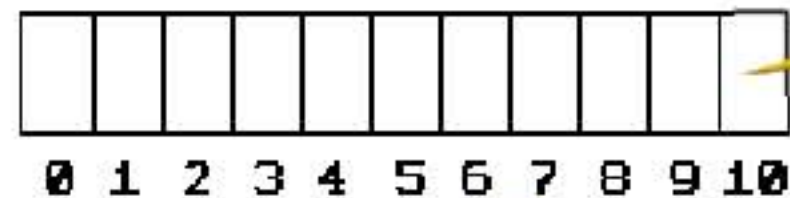
- ▣ Diferentes de outras linguagens, não existe o tipo string
- ▣ Matriz é um tabela $m \times n$ de variáveis de um mesmo tipo
- ▣ Vetor é uma matriz com uma única linha e n colunas de variáveis de um mesmo tipo
- ▣ Declaração do vetor:

```
tipo nome[tamanho];
```

Número de elementos

```
int idade[11];
```

Conteúdo de um único tipo



índice

Vetores e Matrizes

o primeiro índice é 0

o último elemento útil é $n - 1$

Por exemplo, **11** elementos implica elementos de 0 a **10**

Atribuição:

Letra[9] = 'A'

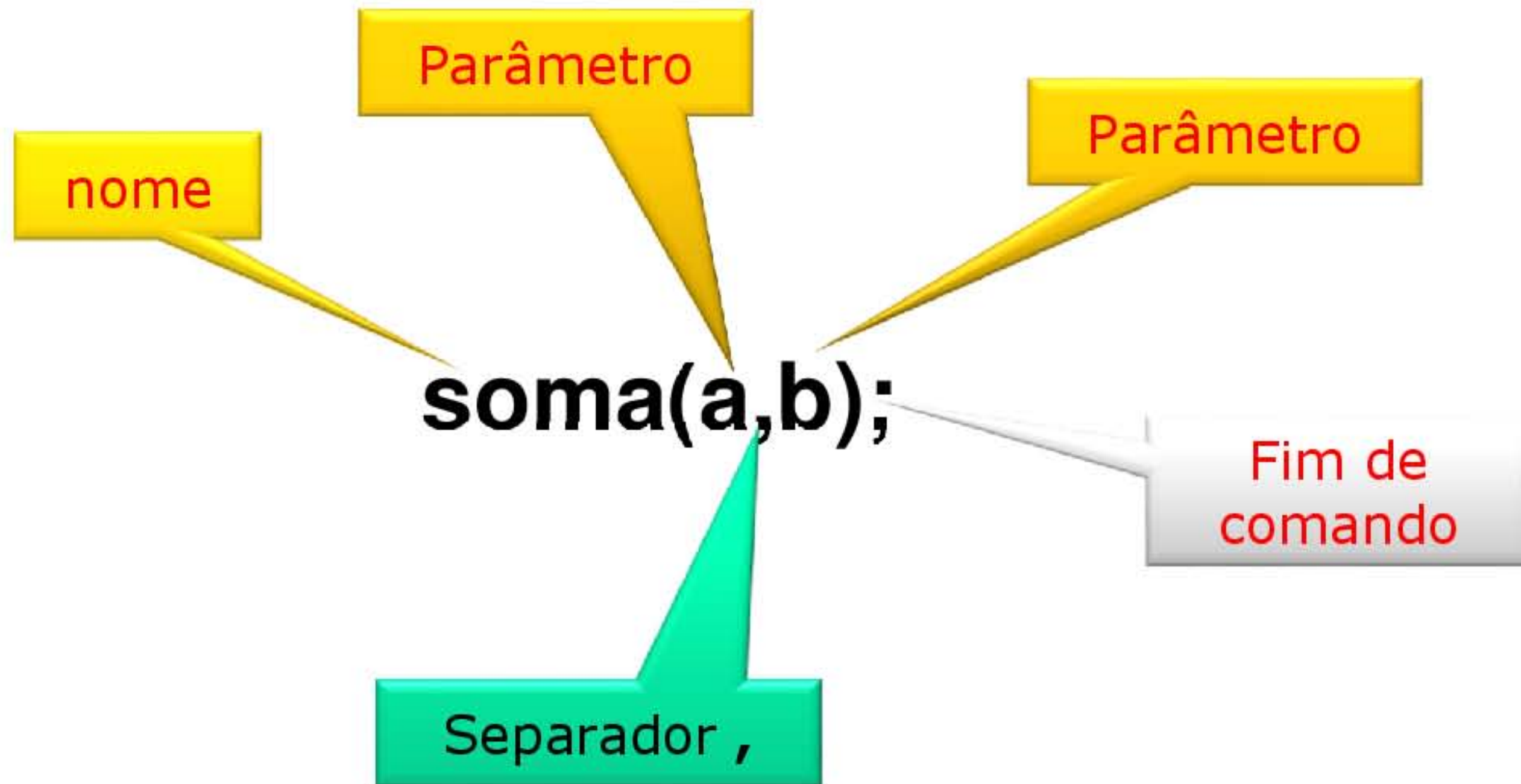
Decisão:

if (Letra[0] == 'C')

A linguagem C não confere o limite do vetor

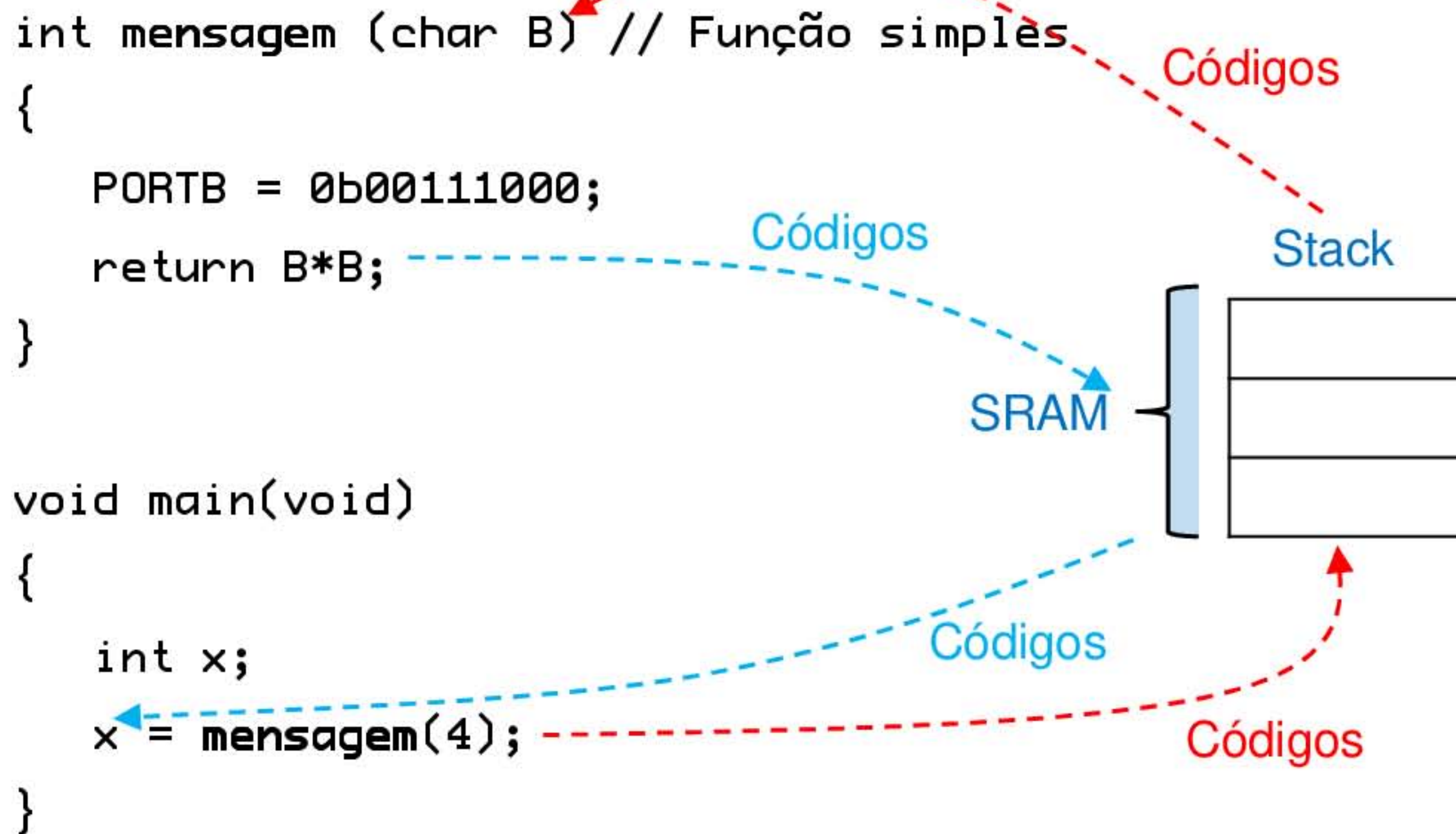
Funções

🖥️ C é formado por funções



Exemplo

- Parâmetros para função são passados pelo stack e a origem normalmente não é alterada.



Sintaxe

O `void` é o tipo vazio utilizado para evitar códigos adicionais de retorno do Stack

<code>void nome_da_função (parâmetros);</code>	Elimina o trânsito de retorno
<code>tipo nome_da_função (void);</code>	Elimina o trânsito de ida dos parâmetros
<code>void nome_da_função (void);</code>	Elimina a ida e a volta de dados

Exemplo

Global

```
int contador;
```

```
int func1(int j)  
{  
    // códigos da função  
}
```

Local

```
void main(void)  
{
```

```
    char condicao;
```

```
    int i;
```

```
    for (i=0; i<100; i=i+1)  
    {
```

```
        contador++;
```

```
        // etc..
```

```
        func1(i);
```

```
    }
```

```
}
```

Exemplo

```
int mult (float a, float b, float c) // Multiplica 3 números
```

```
{  
    int Edge;  
    Edge = a * b * c;  
    PORTC = Edge;  
    return 0;  
}
```

```
void main(void)
```

```
{  
    float x,y;  
    x=23.5;  
    y=12.9;  
    mult (x,y,3.87);  
}
```

☞ $x \rightarrow a$

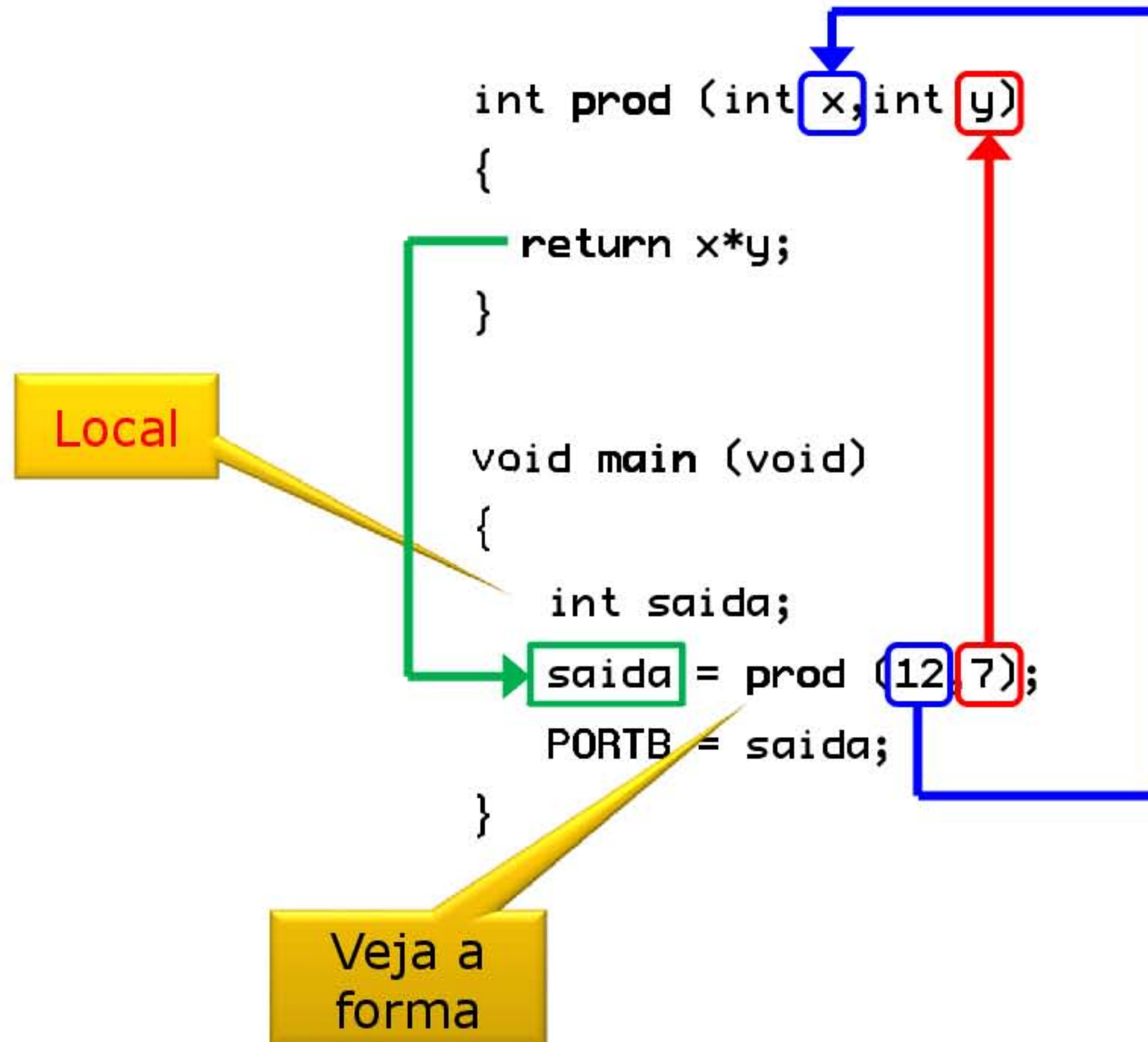
☞ $y \rightarrow b$

☞ $3.87 \rightarrow c$

☞ Não existe retorno para as variáveis originantes pois temos **return 0;** o que as mantém com seus valores originais

☞ E lembre, neste exemplo, que não existe o bit 7 do PORTC

Com retorno de valores





Análise

1. Se não especificamos o tipo de retorno o compilador considera este tipo inteiro – o que não é boa prática.
2. **main** sempre retorna **int**.
3. Idealmente usaremos `return 0;`
4. O tipo de cada receptor deve ser especificado na função.
5. Para microcontroladores usaremos o comando `return [ação];` para retornar um valor



Protótipo de função

- Protótipos são opcionais, mas são exigidos pela sucessora C++.
- Protótipos permitem que C forneça uma verificação mais forte de tipos.

exemplo

```
int prod (int x,int y);
```

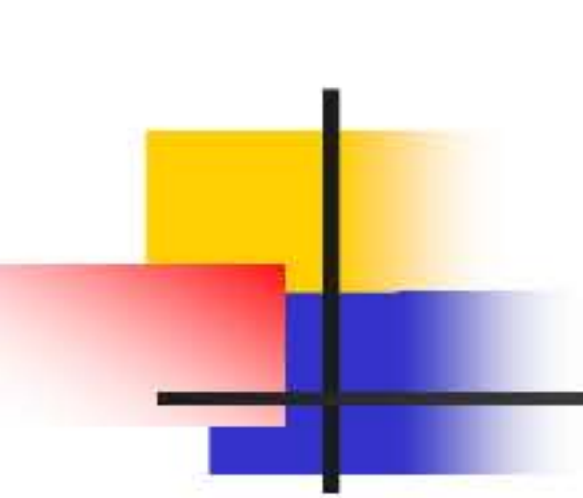
Protótipo

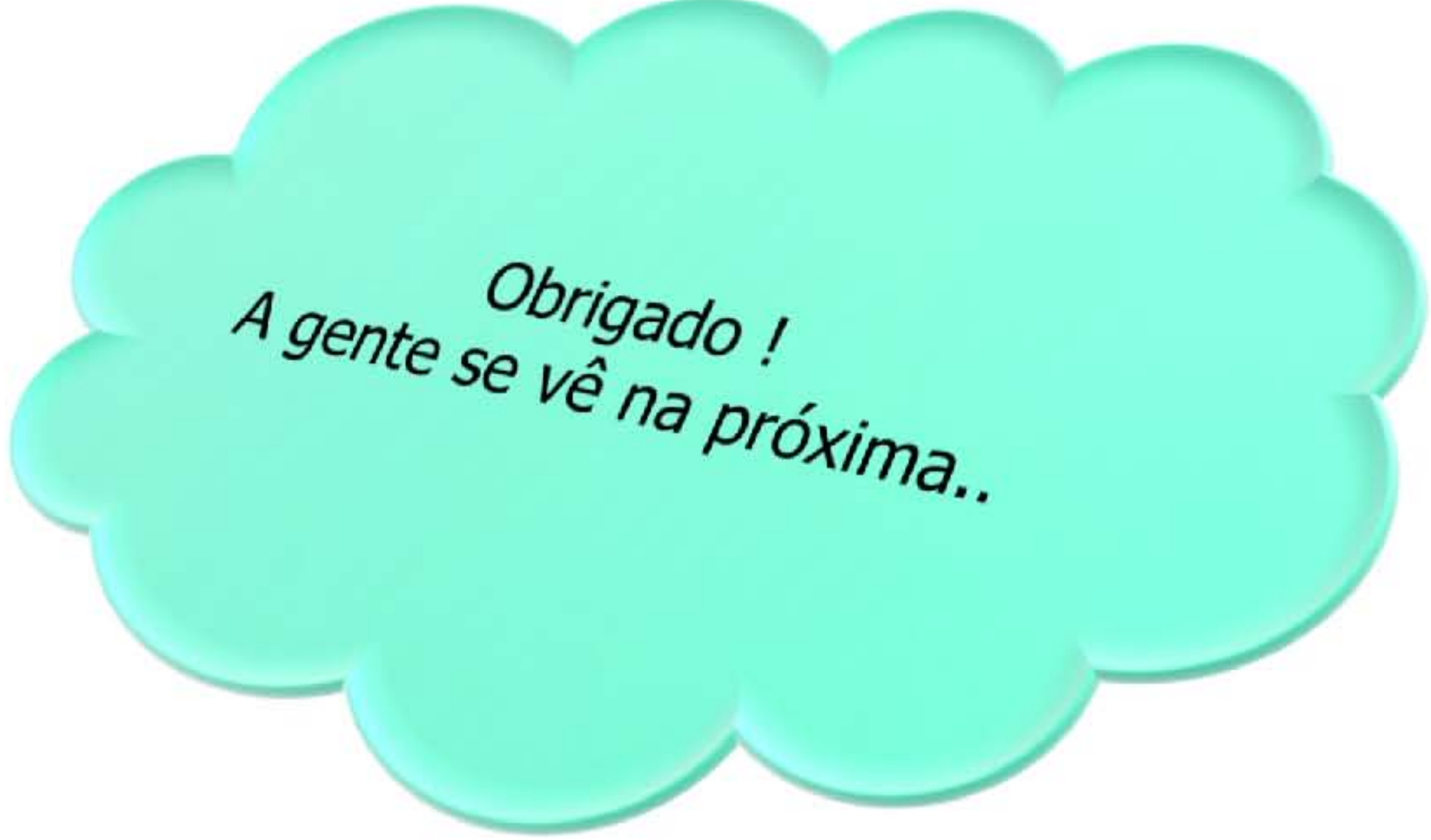
```
void main(void)
{
    int saida;
    saida = prod (12,7);
    PORTD = saida;
}
```

Observe o ':'

```
int prod (int x,int y)
{
    return x+y;
}
```

Código real





Obrigado !
A gente se vê na próxima..