

Algoritmos

Etapas Operacionais; Tipo de Dados; Variáveis.

Prof. Dr. Osmar Betazzi Dordal

2

Etapas Operacionais

- Programa sequencial, como o próprio nome diz, é um conjunto de rotinas programáveis executadas em sequência, ou seja, uma após a outra.
- Podemos organizar essas rotinas em três macroetapas:
 - Entrada de dados (Leitura – ***read***):
 - O computador recebe os dados fornecidos por pessoas ou outros programas.
 - Processamento de dados:
 - Neste momento os **dados recebidos são transformados por meio de operações.**
 - Saída de dados (Escrita – ***write***):
 - Ocorre após haver algum tipo de processamento (novos dados).

Um Primeiro Programa (Algoritmo)

- Programa sequencial, como o próprio nome diz, é um conjunto de rotinas programáveis executadas em sequência, ou seja, uma após a outra.
- **Início e fim do programa:** Linhas 1 e 9;
- **Declarações:** Linhas 2 e 3;
- **Entradas:** Linhas 5 e 6;
- **Processamento:** Linha 7
- **Saída:** Linha 8;

```
1 Program Soma;  
2 var  
3     n1, n2, resultado: integer;  
4 Begin  
5     readln (n1);  
6     readln (n2);  
7     resultado := n1 + n2;  
8     write(resultado);  
9 End.
```

Tipos de Dados

- No **mundo real**, manipulamos diferentes tipos de dados para representar as mais variadas informações;
- Na **computação**, os dados devem ser **definidos** de acordo com **tipos predefinidos pela linguagem** de programação ou estabelecidos pelo programador;
- Tipos primitivos, ou tipos básicos. Eles estão organizados em 3 grupos:
 - Numéricos
 - Caracteres
 - Lógicos

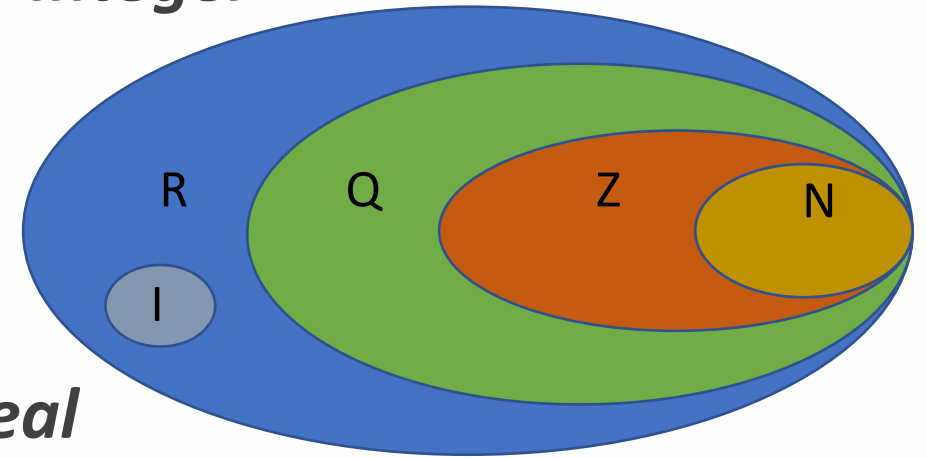
Tipos de Dados

- Dado numérico **Inteiro** (positivo ou negativo) – **Integer**

- Na matemática são conhecidos pelo símbolo \mathbb{Z} ;
- {1; -20; 0; 1500; -5432};
- São números não fracionários.

- Dado numérico **Real** (positivo ou negativo) - **Real**

- Na matemática são conhecidos pelo símbolo \mathbb{R} ;
- {1; -20.4; 556.21; -185.45};
- São números fracionários;
- Todo número inteiro também é real, mas o contrário não é verdade.



Tipos de Dados: Caracteres – *Char*

- Na linguagem **Pascal**, o dado **caractere** costuma ser representado **por um único símbolo** entre aspas simples.
- Pode ser representado por letras maiúsculas ou minúsculas, numerais e símbolos que estão presentes no teclado;
- Exemplos:
 - 'C', 'A', '3', 'b', '-'

Tipos de Dados: Cadeia de Caracteres – *String*

- Na linguagem **Pascal**, o dado *String* costuma ser representado **por um conjunto de símbolos** entre aspas simples.
- Pode ser representado **por um conjunto** de letras maiúsculas ou minúsculas, numerais e símbolos que estão presentes no teclado;
- Exemplos:
 - 'CEP: 80040-540', 'Av. São Luiz, 340', '8.694', 'A', 'b', '-'

Tipos de Dados – Lógico (*Boolean*)

- O dado **lógico**, também conhecido como *booleano*, é o tipo de dado mais simples, pois representa um valor binário, como:
 - 0 ou 1;
 - Sim ou Não;
 - Verdadeiro ou Falso;
 - V ou F.
- No Pascal, os valores aceitos são *True* ou *False*

Variáveis

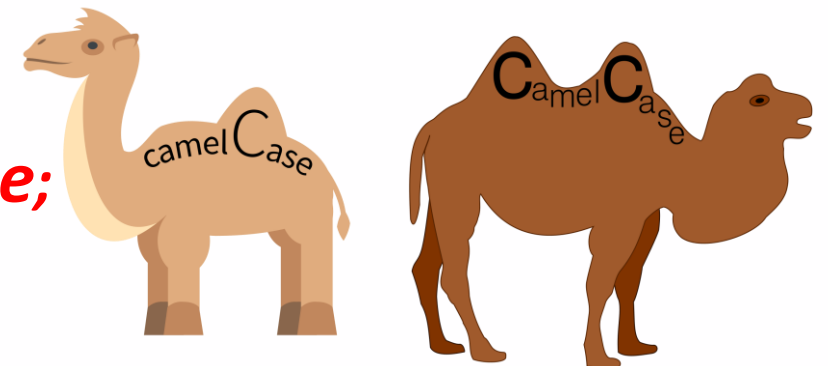
- Variável é uma entidade computacional que **representa um espaço reservado na memória do computador** para armazenamento de dados durante a execução de um programa;
- Esses dados **podem sofrer modificação** ao longo da execução do programa, por isso têm esse nome;
- Para definir as variáveis, é necessário atribuir um nome a elas, conhecido como **identificador**.

Regras de Nomenclatura de Variáveis

- Inicie o nome apenas por caractere alfabético ou “_” (*underline*):
 - Nenhum outro caractere é aceito no início de um identificador (nome de variável).
- Não utilize caracteres especiais na nomenclatura de uma variável:
 - (é, ç, ã, &, %, #, @, *, -, entre outros), exceto o caractere “_” (*underline*).
- Se a variável for identificada por um nome composto, não pode haver espaço em branco:
 - Um modo comum de identificar variáveis com nomes compostos é utilizar o caractere “_” (*underline*).
- **Exemplo:** data_de_nascimento

Regras de Nomenclatura de Variáveis

- Não escolha nomes que já sejam utilizados pela linguagem de programação para representar comandos ou instruções predefinidas:
 - Esses nomes são chamados de **palavras reservadas**.
- Crie identificadores que representem o conteúdo da variável.
- Não utilize o mesmo nome para mais de uma variável.
- ***Camel Case:***
 - Escrevemos: *camelCase* ou *CamelCase*;
 - O padrão que é mais interessante usar é o ***camelCase***;
 - **Exemplo:** dataDeNascimento.



Constantes - *Const*

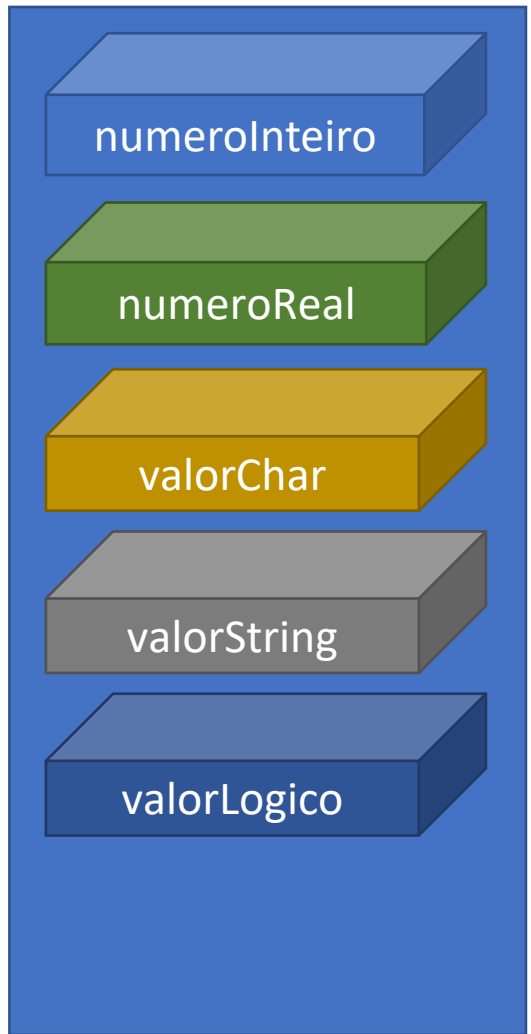
- Além das variáveis, temos as entidades conhecidas como **constantes**, que são a definição de **valores fixos**, estáveis.
- Uma vez atribuído, o **valor de uma constante mantém-se o mesmo até o final da execução do programa.**
- A criação da constante, como um identificador único, utilizamos o comando ***const***. Também é uma convenção declarar uma constante toda em maiúscula
 - Exemplos de Constantes:
 - **ALÍQUOTA_INSS = 0.11, PI = 3.14159**

Declarações das Variáveis e Constantes

Temos no Algoritmo em Pascal todos os tipos de variáveis que foram mencionadas.

```
1 Program Declaracoes ;
2
3 var
4     numeroInteiro: integer;
5     numeroReal: real;
6     valorCaractere: char;
7     valorString: string;
8     valorLogico: boolean;
9
10 Const
11     PI = 3.14159;
12
13 Begin
14
15     numeroInteiro := 1;
16     numeroReal := 1.1;
17     valorCaractere := 'a';
18     valorString := 'abc';
19     valorLogico := true;
20
21     writeln('O valor inteiro é: ', numeroInteiro);
22     writeln('O valor real é: ', numeroReal);
23     writeln('O valor caractere é: ', valorCaractere);
24     writeln('O valor string é: ', valorString);
25     writeln('O valor lógico é: ', valorLogico);
26     writeln('O valor da constante PI é: ', PI);
27
28 End.
```

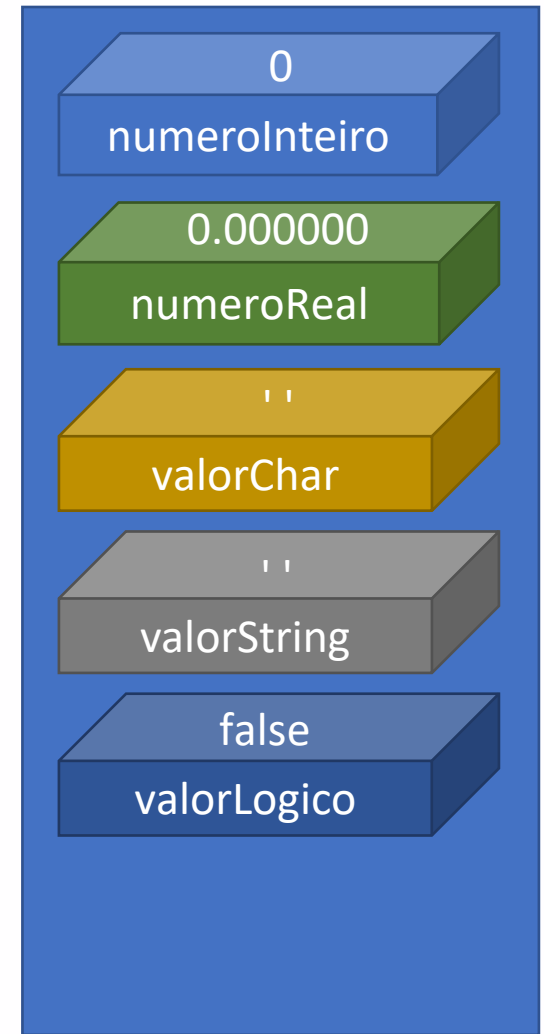
```
1 Program Declaracoes ;
2
3 var
4     numeroInteiro: integer;
5     numeroReal: real;
6     valorCaractere: char;
7     valorString: string;
8     valorLogico: boolean;
9
10 Const
11     PI = 3.14159;
12
13 Begin
14
15     numeroInteiro := 1;
16     numeroReal := 1.1;
17     valorCaractere := 'a';
18     valorString := 'abc';
19     valorLogico := true;
20
21     writeln('O valor inteiro é: ', numeroInteiro);
22     writeln('O valor real é: ', numeroReal);
23     writeln('O valor caractere é: ', valorCaractere);
24     writeln('O valor string é: ', valorString);
25     writeln('O valor lógico é: ', valorLogico);
26     writeln('O valor da constante PI é: ', PI);
27
28 End.
```



```

1 Program Declaracoes ;
2
3 var
4   numeroInteiro: integer;
5   numeroReal: real;
6   valorCaractere: char;
7   valorString: string;
8   valorLogico: boolean;
9
10 Const
11   PI = 3.14159;
12
13 Begin
14
15   numeroInteiro := 1;
16   numeroReal := 1.1;
17   valorCaractere := 'a';
18   valorString := 'abc';
19   valorLogico := true;
20
21   writeln('O valor inteiro é: ', numeroInteiro);
22   writeln('O valor real é: ', numeroReal);
23   writeln('O valor caractere é: ', valorCaractere);
24   writeln('O valor string é: ', valorString);
25   writeln('O valor lógico é: ', valorLogico);
26   writeln('O valor da constante PI é: ', PI);
27
28 End.

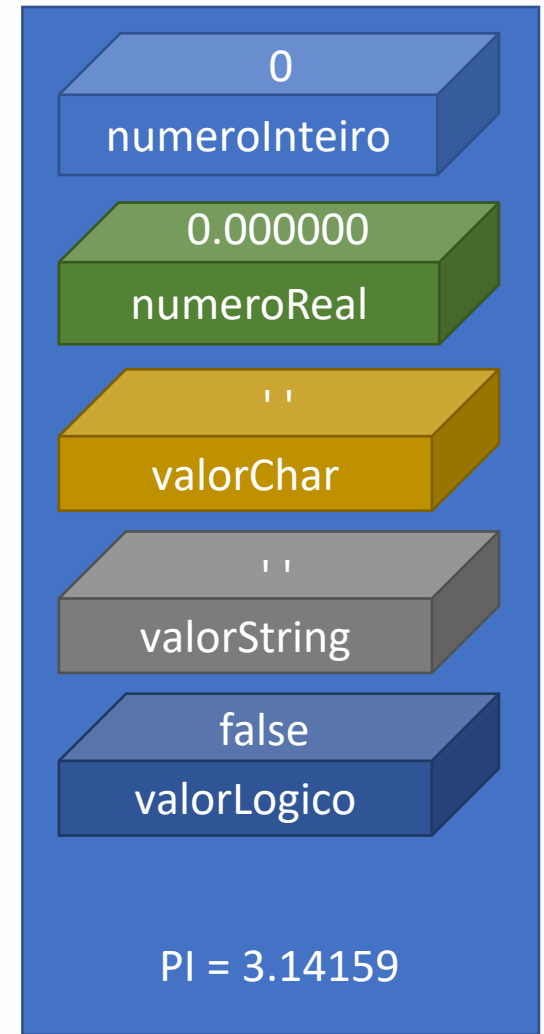
```



```

1 Program Declaracoes ;
2
3 var
4   numeroInteiro: integer;
5   numeroReal: real;
6   valorCaractere: char;
7   valorString: string;
8   valorLogico: boolean;
9
10  Const
11    PI = 3.14159;
12
13  Begin
14
15    numeroInteiro := 1;
16    numeroReal := 1.1;
17    valorCaractere := 'a';
18    valorString := 'abc';
19    valorLogico := true;
20
21    writeln('O valor inteiro é: ', numeroInteiro);
22    writeln('O valor real é: ', numeroReal);
23    writeln('O valor caractere é: ', valorCaractere);
24    writeln('O valor string é: ', valorString);
25    writeln('O valor lógico é: ', valorLogico);
26    writeln('O valor da constante PI é: ', PI);
27
28  End.

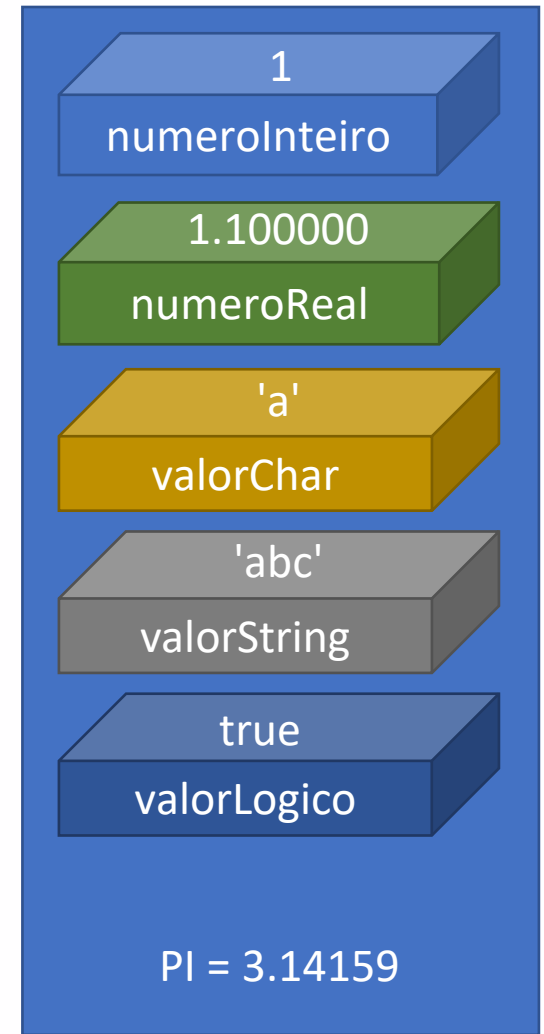
```




```

1 Program Declaracoes ;
2
3 var
4   numeroInteiro: integer;
5   numeroReal: real;
6   valorCaractere: char;
7   valorString: string;
8   valorLogico: boolean;
9
10 Const
11   PI = 3.14159;
12
13 Begin
14
15   numeroInteiro := 1;
16   numeroReal := 1.1;
17   valorCaractere := 'a';
18   valorString := 'abc';
19   valorLogico := true;
20
21   writeln('O valor inteiro é: ', numeroInteiro);
22   writeln('O valor real é: ', numeroReal);
23   writeln('O valor caractere é: ', valorCaractere);
24   writeln('O valor string é: ', valorString);
25   writeln('O valor lógico é: ', valorLogico);
26   writeln('O valor da constante PI é: ', PI);
27
28 End.

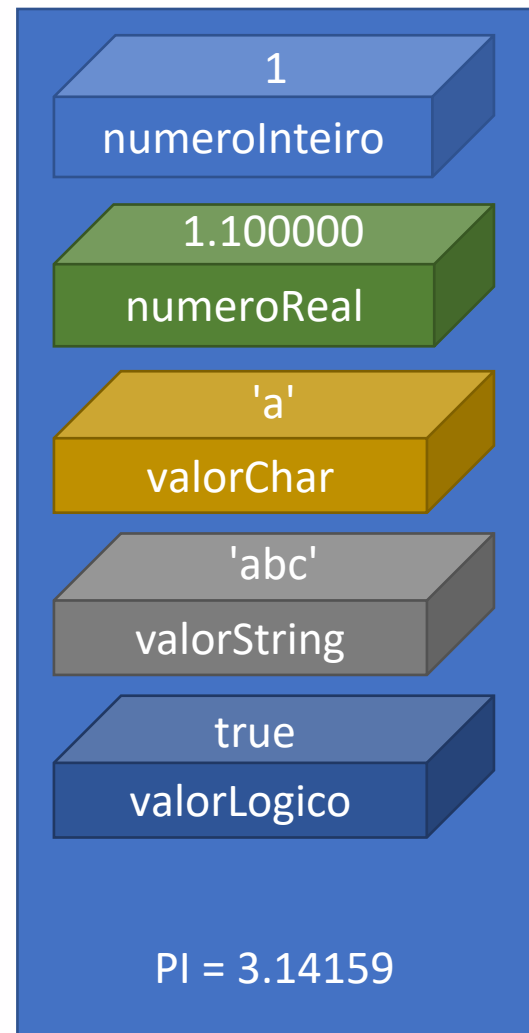
```



```

1 Program Declaracoes ;
2
3 var
4     numeroInteiro: integer;
5     numeroReal: real;
6     valorCaractere: char;
7     valorString: string;
8     valorLogico: boolean;
9
10 Const
11     PI = 3.14159;
12
13 Begin
14
15     numeroInteiro := 1;
16     numeroReal := 1.1;
17     valorCaractere := 'a';
18     valorString := 'abc';
19     valorLogico := true;
20
21     writeln('O valor inteiro é: ', numeroInteiro);
22     writeln('O valor real é: ', numeroReal);
23     writeln('O valor caractere é: ', valorCaractere);
24     writeln('O valor string é: ', valorString);
25     writeln('O valor lógico é: ', valorLogico);
26     writeln('O valor da constante PI é: ', PI);
27
28 End.

```



```
1 Program Declaracoes ;
2
3 var
4   numeroInteiro: integer;
5   numeroReal: real;
6   valorCaractere: char;
7   valorString: string;
8   valorLogico: boolean;
```

```
10 Const
```

```
11   PI = 3.14159;
```

```
13 Begin
```

```
15   numeroInteiro := 1;
16   numeroReal := 1.1;
17   valorCaractere := 'a';
18   valorString := 'abc';
19   valorLogico := true;
```

```
21   writeln('O valor inteiro é: ', numeroInteiro);
22   writeln('O valor real é: ', numeroReal);
23   writeln('O valor caractere é: ', valorCaractere);
24   writeln('O valor string é: ', valorString);
25   writeln('O valor lógico é: ', valorLogico);
26   writeln('O valor da constante PI é: ', PI);
```

```
28 End.
```

O valor inteiro é:

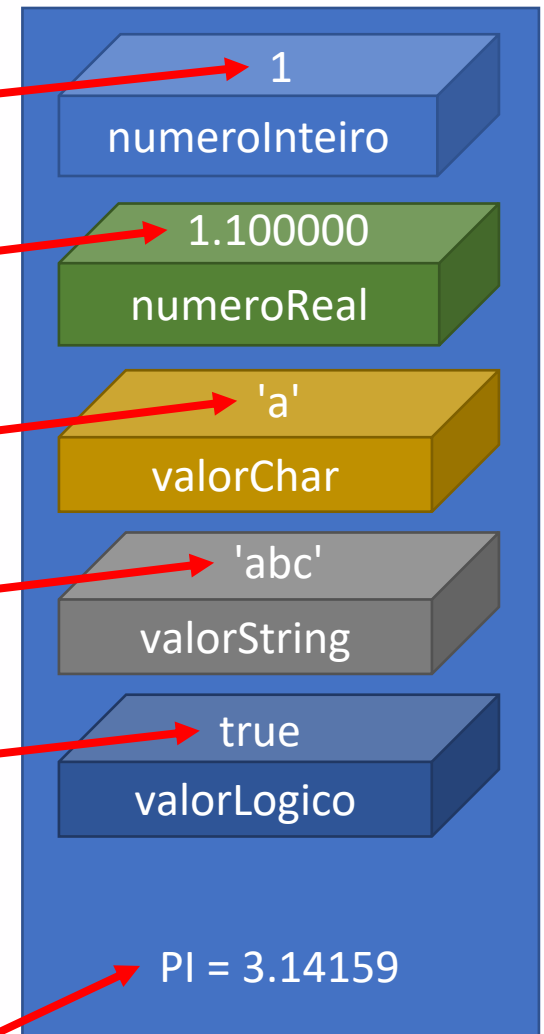
O valor real é:

O valor caractere é:

O valor *string* é:

O valor lógico é:

O valor da constante PI é:



Pascal vs Python

```
1 Program Soma;  
2 var  
3   n1, n2, resultado: integer;  
4 Begin  
5   readln(n1);  
6   readln(n2);  
7   resultado := n1 + n2;  
8   write(resultado);  
9 End.
```

```
n1 = input()  
n2 = input()  
resultado = n1 + n2  
print(resultado)
```

Escreva o código em Python e verifique a saída:

Mas antes, vamos ajudar o usuário do algoritmo em Pascal...

Pascal vs Python

```
1 Program Soma;  
2 var  
3   n1, n2, resultado: integer;  
4 Begin  
5   write('Informe um valor inteiro:');  
6   readln(n1);  
7   write('Informe um valor inteiro:');  
8   readln(n2);  
9   resultado := n1 + n2;  
10  write('O resultado é:', resultado);  
11 End.
```

Qual o resultado em
ambos os algoritmos?

```
1 n1 = input("informe o primeiro valor [número inteiro]:")  
2 n2 = input("informe o segundo valor [número inteiro]:")  
3 resultado = n1 + n2  
4 print("O resultado é:", resultado)
```

Pascal vs Python

```
1 Program Soma;  
2 var  
3   n1, n2, resultado: integer;  
4 Begin  
5   write('Informe um valor inteiro:');  
6   readln(n1);  
7   write('Informe um valor inteiro:');  
8   readln(n2);  
9   resultado := n1 + n2;  
10  write('O resultado é:', resultado);  
11 End.
```

E agora?

```
1 n1 = int(input("informe o primeiro valor [número inteiro]:"))  
2 n2 = int(input("informe o segundo valor [número inteiro]:"))  
3 resultado = n1 + n2  
4 print("O resultado é:", resultado)
```

Pascal vs Python

```
1 n1 = input("informe o primeiro valor [número inteiro]:")
2 n2 = input("informe o segundo valor [número inteiro]:")
3 resultado = n1 + n2
4 print("O resultado é:", resultado)
```

```
1 n1 = int(input("informe o primeiro valor [número inteiro]:"))
2 n2 = int(input("informe o segundo valor [número inteiro]:"))
3 resultado = n1 + n2
4 print("O resultado é:", resultado)
```

```
1 n1 = input("informe o primeiro valor [número inteiro]:")
2 n2 = input("informe o segundo valor [número inteiro]:")
3 resultado = int(n1) + int(n2)
4 print("O resultado é:", resultado)
```

Referências

- FORBELLONE, André L. **Lógica de Programação**. Prentice Hall Brasil, 3ª edição, 2005.
- VELOSO, Paulo; et al. **Estrutura de dados**. Rio de Janeiro: Campus, 4ª edição, 1996.
- LAGES & GUIMARAES. **Algoritmos e Estrutura de dados**. Ed. LTC, 1994.
- FARRER, H. **Algoritmos estruturados**. Rio de Janeiro: Guanabara Koogan, 3ª edição, 1989.
- LUIZ, Jaime. **Estrutura de dados e seus algoritmos**. Editora LTC.
- GUEDES, S. **Lógica de Programação Algorítmica**. Editora Pearson, 2014.
- MANZANO, José Augusto N. G. **Algoritmos lógica para desenvolvimento de programação de computadores**. Ed. 1, São Paulo, Erica 2016.
- MANZANO, José Augusto N. G. **Algoritmos técnicas de programação**. Ed 2, São Paulo, Erica, 2016.