

# Introdução à Programação C++ para iniciantes e programadores de primeira viagem – Parte 1

Uberlândia, 21 de outubro de 2014

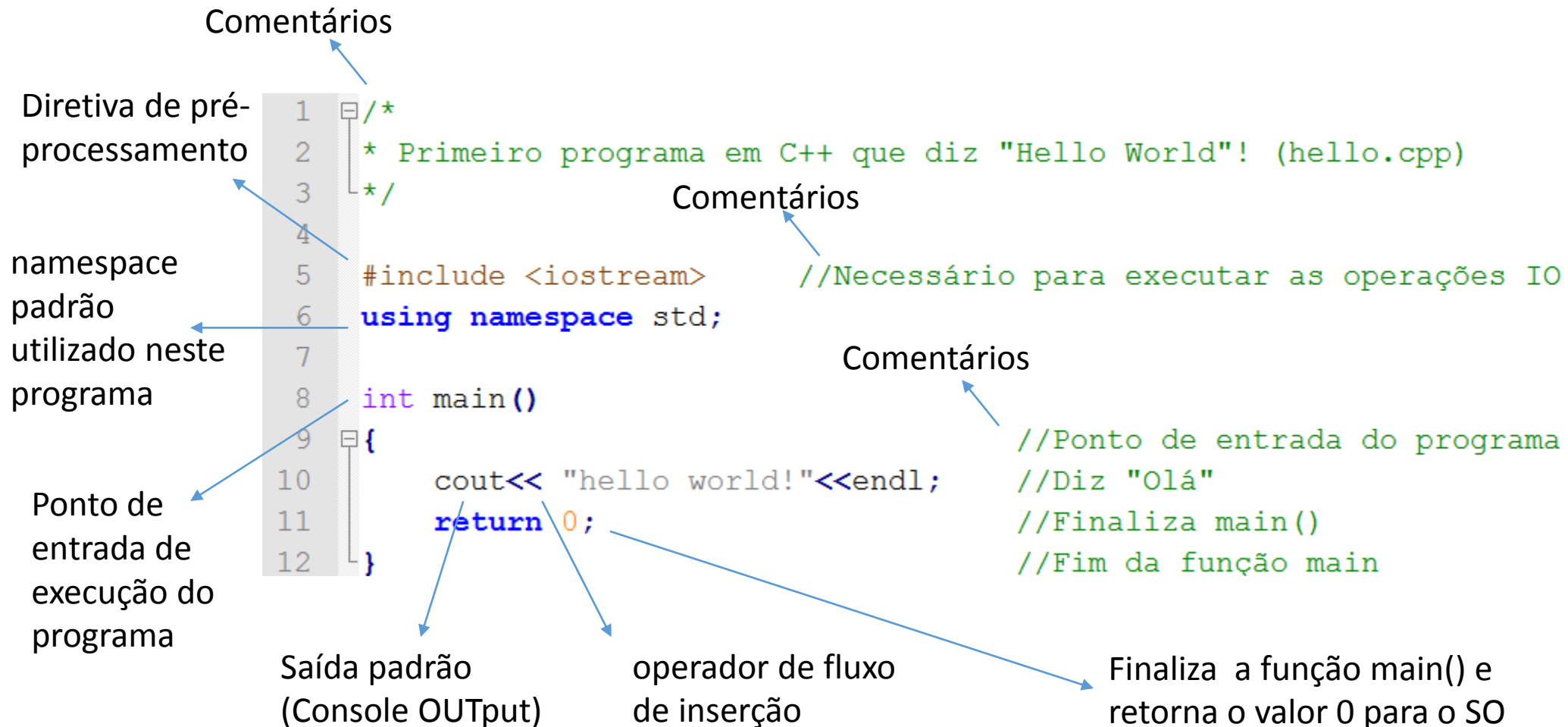
Daniel Teodoro Gonçalves Mariano  
dtgmario@gmail.com

# Objetivos do minicurso

- Why programming is important?
  - <http://www.youtube.com/watch?v=Dv7gLpW91DM>

# 1. Vamos começar?

Escreva o seu primeiro programa “Hello world!”



# Agora escreva a sua versão de “Hello world” !

Utilize o comando “cout” para imprimir alguma mensagem na tela.

- “Olá, meu nome é João.”
- “Está fazendo um calor insuportável não é mesmo? ”

Você é o artista do seu programa, faça da forma que achar melhor!

USE COMENTÁRIOS!!!

UTILIZE ESPAÇOS BRANCOS  
ADICIONAIS E NOVAS LINHAS!!!

## 2. C++: Terminologia e sintaxe

Uma declaração realiza uma parte de ação de programação. Deve ser finalizada por um ponto e vírgula ;

Statement

Um bloco é um grupo de instruções de programação delimitados por chaves {}

Preprocessor  
Directive

Block

**#include** é uma diretiva de pré-processamento (não é uma declaração de programação). É processado antes de compilar o programa. (Começa com **#** e **não** é finalizada com ;)

a ROSA ≠ a Rosa ≠ a rosa

Declarações não-executáveis,  
sendo ignoradas pelo compilador

Comments

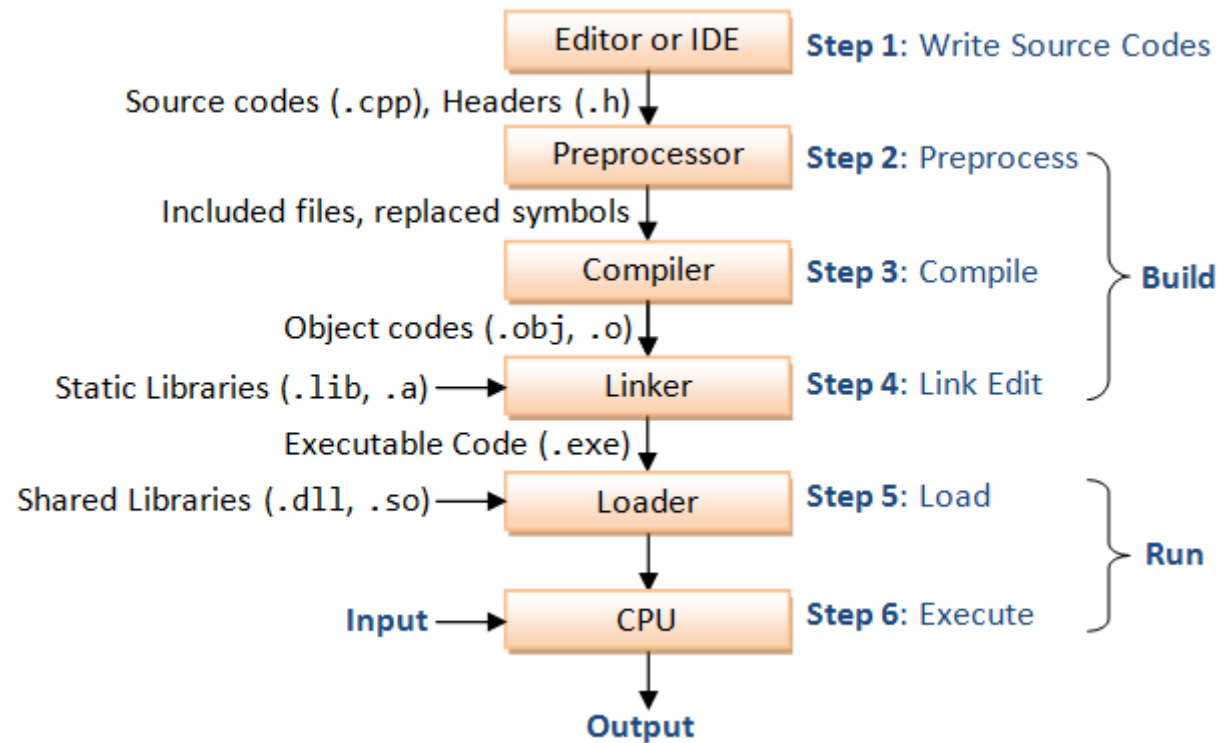
n-linhas: /\* ... \*/  
1-linha: // ...

Whitespaces

Case  
Sensitivity

Blanks, tabulações e linhas novas são chamadas coletivamente de espaços em branco. Espaços em branco extra são ignorados. São utilizados para uma melhor compreensão do programa.

### 3. O processo de escrever um programa em C++



## 4. Um modelo de programa em C++

```
1  /*
2   * Comentário para indicar a finalidade deste programa (filename.cpp)
3   */
4  #include <iostream>
5  using namespace std;
6
7  int main() {
8      // Suas declarações de programação AQUI!
9
10     return 0;
11 }
```

## 5. Output via “cout <<” (Console **OUT**put)

```
cout << "hello" << " world, " << "again!" << endl;  
cout << "hello," << endl << "one more time. " << endl << 5 << 4 << 3 << " " << 2.2 << " " << 1.1 << endl;
```

```
hello world, again!  
hello,  
one more time.  
543 2.2 1.1  
—
```

```
cout << "hello world, again!\n";  
cout << "\thello,\none\tmore\ttime.\n";
```

```
hello world, again!  
      hello,  
      one   more   time.  
—
```

Eu recomendo fortemente que você use **endl** para imprimir uma nova linha, em vez de '\n'. Isto porque o delimitador de linha é dependente do sistema:

Windows => "\r\n";

UNIX / Linux / Mac => '\n'.

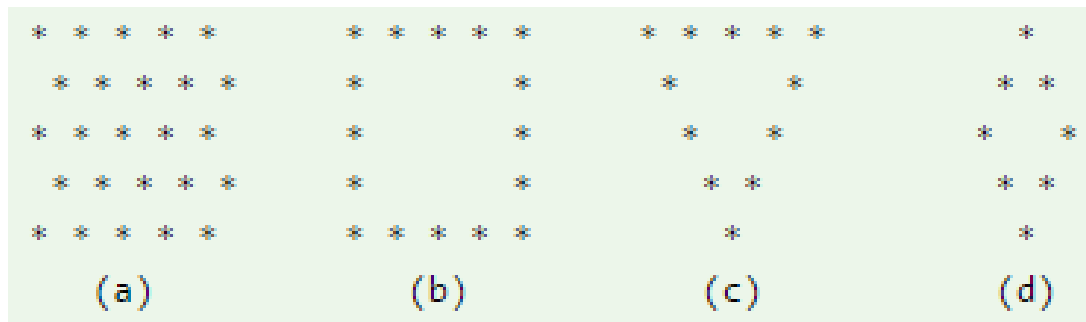
O endl produz nova linha específica do sistema.

Além disso, endl garante que a saída é liberado; enquanto '\n' não.



# Exercício

- 1) Escreve um programa chamado PrintPatternX.cpp para cada um dos padrões seguintes.
  - Utilize “**cout <<**” para cada linha de saída.
  - Finalize cada linha imprimindo “**endl**”.



- 2) Você é capaz de escrever o padrão acima utilizando uma única declaração “**cout <<**” ?

## 6. Input via “cin >>” (**C**onsole **I**Nput )

“cout” permite enviar uma informação para o usuário!

Como o meu programa poderia receber uma informação do meu usuário???

```

1  /*
2   * Prompt user for two integers and print their sum, difference, product and quotient
3   * (IntegerArithmetic.cpp)
4   */
5  #include <iostream>
6  using namespace std;
7
8  int main() {
9      int firstInt; // Declare a variable named firstInt of the type int (integer)
10     int secondInt; // Declare a variable named secondInt of the type int
11     int sum, difference, product, quotient;
12     // Declare 4 variables of the type int to keep the results
13
14     cout << "Enter first integer: "; // Display a prompting message
15     cin >> firstInt; // Read input from keyboard (cin) into firstInt
16     cout << "Enter second integer: "; // Display a prompting message
17     cin >> secondInt; // Read input into secondInt
18
19     // Perform arithmetic operations
20     sum = firstInt + secondInt;
21     difference = firstInt - secondInt;
22     product = firstInt * secondInt;
23     quotient = firstInt / secondInt;
24
25     // Print the results
26     cout << "The sum is: " << sum << endl;
27     cout << "The difference is: " << difference << endl;
28     cout << "The product is: " << product << endl;
29     cout << "The quotient is: " << quotient << endl;
30
31     return 0;
32 }

```

Declaramos algumas variáveis int (inteiro).

Uma variável é um local de armazenamento nomeado que pode armazenar um valor de tipo de dado particular (nessa caso int)

OUTPUT

INPUT

Operadores aritméticos

```

1  /*
2   * Prompt user for two integers and print their sum, difference, product and quotient
3   * (IntegerArithmetic.cpp)
4   */
5  #include <iostream>
6  using namespace std;
7
8  int main() {
9      int firstInt; // Declare a variable named firstInt of the type int (integer)
10     int secondInt; // Declare a variable named secondInt of the type int
11     int sum, difference, product, quotient;
12     // Declare 4 variables of the type int to keep the results
13
14     cout << "Enter first integer: "; // Display a prompting message
15     cin >> firstInt; // Read input from keyboard (cin) into firstInt
16     cout << "Enter second integer: "; // Display a prompting message
17     cin >> secondInt; // Read input into secondInt
18
19     // Perform arithmetic operations
20     sum = firstInt + secondInt;
21     difference = firstInt - secondInt;
22     product = firstInt * secondInt;
23     quotient = firstInt / secondInt;
24
25     // Print the results
26     cout << "The sum is: " << sum << endl;
27     cout << "The difference is: " << difference << endl;
28     cout << "The product is: " << product << endl;
29     cout << "The quotient is: " << quotient << endl;
30
31     return 0;
32 }

```


```

Enter first integer: 99
Enter second integer: 4
The sum is: 103
The difference is: 95
The product is: 396
The quotient is: 24

```

# Lendo diversos itens com um único **cin**

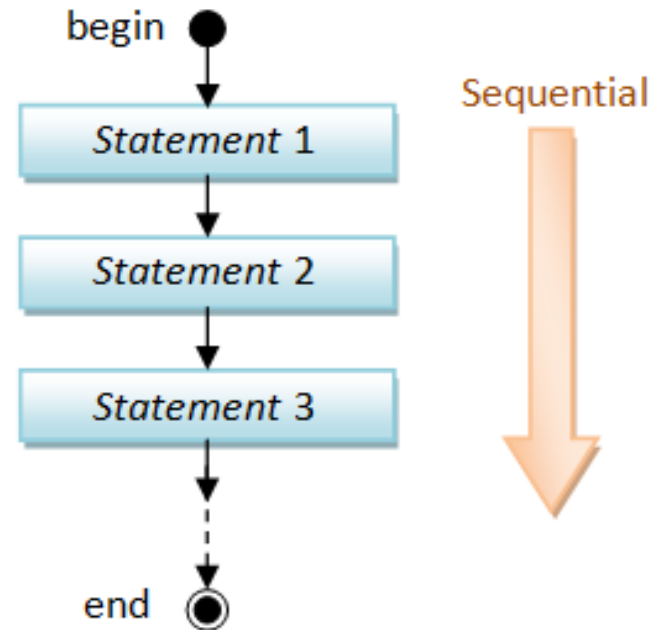
```
cout << "Enter two integers (separated by space): "; // Put out a prompting message
cin >> firstInt >> secondInt; // Read two values into respective variables
sum = firstInt + secondInt;
cout << "The sum is: " << sum << endl;
```



```
Enter two integers: 77 88
The sum of 77 and 88 is 165
```

## 7. O que é um programa ?

Um programa é uma sequência de instruções (chamadas instruções de programação) executada uma após a outra. Acontece de forma sequencial geralmente.



```

1  /*
2   * Prompt user for the radius of a circle and compute its area and circumference
3   * (CircleComputation.cpp)
4   */
5  #include <iostream>
6  using namespace std;
7
8  int main() {
9      double radius, circumference, area; // Declare 3 floating-point variables
10     const double PI = 3.14159265; // Declare and define PI
11
12     cout << "Enter the radius: "; // Prompting message
13     cin >> radius; // Read input into variable radius
14
15     // Compute area and circumference
16     area = radius * radius * PI;
17     circumference = 2.0 * radius * PI;
18
19     // Print the results
20     cout << "The radius is: " << radius << endl;
21     cout << "The area is: " << area << endl;
22     cout << "The circumference is: " << circumference << endl;
23
24     return 0;
25 }

```

Declaramos três variáveis chamadas radius, circumference, area. Uma variável double, ao contrário da int, pode guardar números reais (ponto flutuante).

Declaramos uma variável do tipo double “constante”, chamada PI, cujo valor foi inicializado em 3.14159265 e não pode ser mudado

OUTPUT  
e  
INPUT

Executa os cálculos

Impressão dos  
resultados

```

1  /*
2   * Prompt user for the radius of a circle and compute its area and circumference
3   * (CircleComputation.cpp)
4   */
5  #include <iostream>
6  using namespace std;
7
8  int main() {
9      double radius, circumference, area; // Declare 3 floating-point variables
10     const double PI = 3.14159265;      // Declare and define PI
11
12     cout << "Enter the radius: "; // Prompting message
13     cin >> radius;                // Read input into variable radius
14
15     // Compute area and circumference
16     area = radius * radius * PI;
17     circumference = 2.0 * radius * PI;
18
19     // Print the results
20     cout << "The radius is: " << radius << endl;
21     cout << "The area is: " << area << endl;
22     cout << "The circumference is: " << circumference << endl;
23
24     return 0;
25 }

```

Qual é a saída desse programa?

```

Enter the radius: 1.2
The radius is: 1.2
The area is: 4.52389
The circumference is: 7.53982

```



## 8. O que é uma variável?

Programas de computador manipulam (ou processam) informações (data).

Uma variável é usada para armazenar um pedaço de informação para processamento.

É chamada de variável pois seu valor armazenado pode ser alterado.

NAME	VALUE	TYPE
number	123	int
sum	-456	int
pi	3.1416	double
average	-55.66	double

Uma variável possui um **nome**, armazena um **valor** de um **tipo** declarado

# Declarando uma variável

// Syntax: Declare a variable of a type

`var-type` `var-name`;

// Example:

`int sum;`  
`double radius;`

// Syntax: Declare multiple variables of the same type

`var-type` `var-name-1, var-name-2, ...;`

// Example:

`int sum, difference, product, quotient;`  
`double area, circumference;`

// Syntax: Declare a variable of a type, and assign an initial value

`var-type` `var-name` = `initial-value`;

// Example:

`int sum = 0;`  
`double pi = 3.14159265;`

// Syntax: Declare multiple variables of the same type with initial values

`var-type` `var-name-1 = initial-value-1, var-name-2 = initial-value-2, ... ;`

// Example:

`int firstNumber = 1, secondNumber = 2;`

# Exemplos de declaração

Cada variável só pode ser declarada uma única vez

Voce pode declarar uma variável em qualquer lugar dentro do programa contanto que ela seja declarada antes de ser utilizada.

```
int number;           // Declare a variable named "number" of the type "int" (integer)
number = 99;          // Assign an integer value of 99 to the variable "number"
number = 88;          // Re-assign a value of 88 to "number"
number = number + 1;  // Evaluate "number + 1", and assign the result back to "number"
int sum = 0;          // Declare an int variable named sum and assign an initial value of 0
sum = sum + number;   // Evaluate "sum + number", and assign the result back to "sum", i.e. add number into sum
int num1 = 5, num2 = 6; // Declare and initialize two int variables in one statement, separated by a comma
double radius = 1.5;  // Declare a variable name radius, and initialize to 1.5
int number;           // ERROR: A variable named "number" has already been declared
sum = 55.66;          // WARNING: The variable "sum" is an int. It shall not be assigned a floating-point number
sum = "Hello";        // ERROR: The variable "sum" is an int. It cannot be assigned a text string
```

Uma vez que o tipo da variável é declarado, ela poderá armazenar valores pertencendo à aquele tipo particular.

Por exemplo uma variável int pode armazenar um valor inteiro como 123 e não pode armazenar um valor ponto flutuante como -2.17 ou um texto como "Hello". O tipo da variável não pode ser alterado dentro do programa.

# Qual a diferença entre atribuição e igualdade?

**Atribuição (=)** em **programação** é **diferente** de **igualdade** matemática

$X = X + 1$   ATRIBUIÇÃO

$X + Y = 1$   IGUALDADE

Em **programação** a **igualdade** matemática é representado por **==**  
Veremos isso de forma detalhada mais adiante!

## 9. Operações aritméticas básicas

Operador	Significado	Exemplo
+	Adição	$x + y$
-	Subtração	$x - y$
*	Multiplicação	$x * y$
/	Divisão	$x / y$
%	Módulo (Resto)	$x \% y$
++	Incremento por 1	++x ou x++
--	Decremento por 1	--x ou x--

# Exemplo

```
1  /*
2   * Test arithmetic operations (TestArithmetics.cpp)
3   */
4  #include <iostream>
5  using namespace std;
6
7  int main() {
8
9      int number1, number2; // Declare 2 integer variable number1 and number2
10     int sum, difference, product, quotient, remainder; // declare 5 int variables
11
12     // Prompt user for the two numbers
13     cout << "Enter two integers (separated by space): ";
14     cin >> number1 >> number2;
15
16     // Do arithmetic Operations
17     sum = number1 + number2;
18     difference = number1 - number2;
19     product = number1 * number2;
20     quotient = number1 / number2;
21     remainder = number1 % number2;
22 }
```

```

23     cout << "The sum, difference, product, quotient and remainder of "
24         << number1 << " and " << number2 << " are "
25         << sum << ", "
26         << difference << ", "
27         << product << ", "
28         << quotient << ", and "
29         << remainder << endl;
30
31     // Increment and Decrement
32     ++number1;    // Increment the value stored in variable number1 by 1
33                 // same as "number1 = number1 + 1"
34     --number2;    // Decrement the value stored in variable number2 by 1
35                 // same as "number2 = number2 - 1"
36     cout << "number1 after increment is " << number1 << endl;
37     cout << "number2 after decrement is " << number2 << endl;
38
39     quotient = number1 / number2;
40     cout << "The new quotient of " << number1 << " and " << number2
41         << " is " << quotient << endl;
42
43     return 0;
44 }

```

Qual é a saída desse programa?

```

Enter two integers: 98 5
The sum, difference, product, quotient and remainder of 98 and 5 are 103, 93, 49
0, 19, and 3
number1 after increment is 99
number2 after decrement is 4
The new quotient of 99 and 4 is 24

```

# 10. E se você precisar somar 1000 números?

E se você precisar escrever um programa que necessite somar 1000 números?

Como vocês implementariam?

Você estaria disposto a escrever mais de 1000 linhas de código?

Escrever 1000 linhas de código é uma opção, mas ela é a melhor forma?