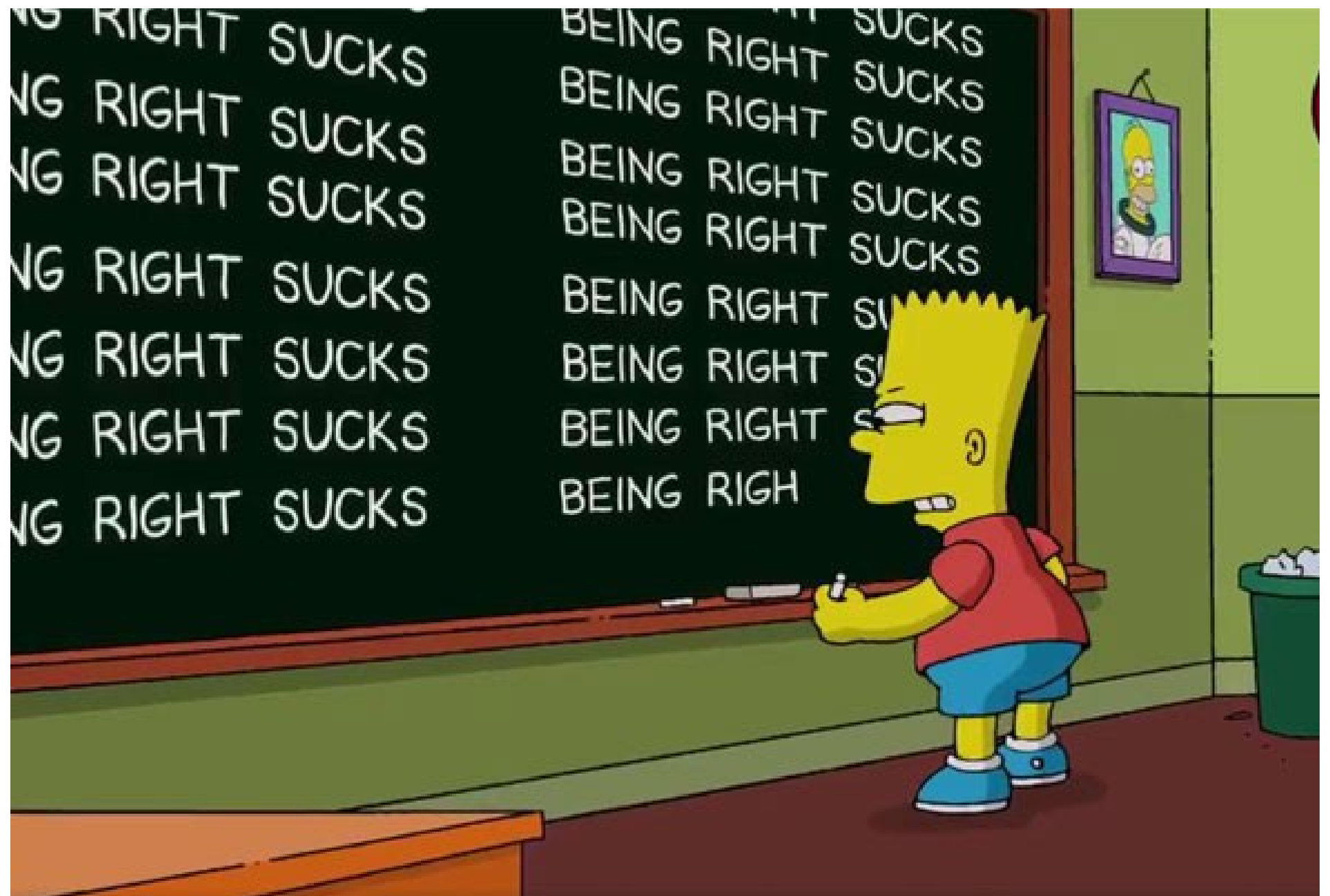


# UNIDADE II: ESTRUTURAS DE REPETIÇÃO

- Como imprimir os 1000 primeiros números a partir do 1?



- Como imprimir os 1000 primeiros números a partir do 1?

- Uma solução seria:

```
escrever: 1;
```

```
...
```

```
escrever: 1000;
```

- Como imprimir os 1000 primeiros números a partir do 1?

- Uma solução seria:

```
escrever: 1;  
  
...  
  
escrever: 1000;
```

**A solução acima não é prática**

- Como imprimir os 1000 primeiros números a partir do 1?
- Outra solução seria utilizar o comando enquanto (*while*)

# Comando Enquanto

**enquanto** (expressão) **faça**

|

lista de comandos

**fim enquanto**

# *while*: Comando Enquanto em C-like

```
while (expressão) {
```

```
    lista de comandos
```

```
}
```



- Como imprimir os 1000 primeiros números a partir do 1?
- Outra solução seria utilizar o comando enquanto (*while*)

```
int i = 1;  
  
while (i <= 1000){  
    escrever: i;  
    i++;  
}
```

## Exercício

● Como imprimir os 1000 primeiros números a partir do 1?

● Outra solução seria utilizar o comando enquanto (*while*)

```
int i = 1;  
  
while (i <= 1000){  
    escrever: i;  
    i++;  
}
```

TELA

QUADRO DE MEMÓRIA	
i	

## Exercício

● Como imprimir os 1000 primeiros números a partir do 1?

● Outra solução seria utilizar o comando enquanto (*while*)

```
int i = 1;
```

```
while (i <= 1000){  
    escrever: i;  
    i++;  
}
```

TELA
1
2
3
...
1000

QUADRO DE MEMÓRIA									
i	1	2	3	4	5	6	7	8	9

## Exercício

● Como imprimir os 1000 primeiros números a partir do 1?

● Outra solução seria utilizar o comando enquanto (*while*)

```
int i = 1;
```

```
while (i <= 1000){
```

```
    escrever: i;
```

```
    i++;
```

```
}
```

TELA
1
2
3
...
1000

QUADRO DE MEMÓRIA									
i	1	2	3	4	5	6	7	8	9

## Exercício

● Como imprimir os 1000 primeiros números a partir do 1?

● Outra solução seria utilizar o comando enquanto (*while*)

```
int i = 1;
```

```
while (i <= 1000){
```

**true**

```
    escrever: i;
```

```
    i++;
```

```
}
```

TELA
1
2
3
...
1000

QUADRO DE MEMÓRIA									
i	1	2	3	4	5	6	7	8	9

## Exercício

- Como imprimir os 1000 primeiros números a partir do 1?

- Outra solução seria utilizar o comando enquanto (*while*)

```
int i = 1;  
  
while (i <= 1000){  
    escrever: i;  
    i++;  
}
```

TELA

1

2

3

...

1000

QUADRO DE MEMÓRIA

i

1

2

3

4

5

6

7

8

9

## Exercício

● Como imprimir os 1000 primeiros números a partir do 1?

● Outra solução seria utilizar o comando enquanto (*while*)

```
int i = 1;  
  
while (i <= 1000){  
    escrever: i;  
    i++;  
}
```

TELA
1
2
3
...
1000

QUADRO DE MEMÓRIA									
i	1	2	3	4	5	6	7	8	9

● Como imprimir os 1000 primeiros números a partir do 1?

● Outra solução seria utilizar o comando enquanto (*while*)

```
int i = 1;
```

```
while (i <= 1000){ true
```

```
    escrever: i;
```

```
    i++;
```

```
}
```

TELA
1
2
3
...
1000

QUADRO DE MEMÓRIA									
i	1	2	3	4	5	6	7	8	9



- Como imprimir os 1000 primeiros números a partir do 1?

- Outra solução seria utilizar o comando enquanto (*while*)

```
int i = 1;  
  
while (i <= 1000){  
    escrever: i;  
    i++;  
}
```

TELA
1
2
3
...
1000

QUADRO DE MEMÓRIA									
i	1	2	3	4	5	6	7	8	9

## Exercício

- Como imprimir os 1000 primeiros números a partir do 1?

- Outra solução seria utilizar o comando enquanto (*while*)

```
int i = 1;  
  
while (i <= 1000){  
    escrever: i;  
    i++;  
}
```

TELA

1

2

3

...

1000

QUADRO DE MEMÓRIA

i

1

2

3

4

5

6

7

8

9

- Como imprimir os 1000 primeiros números a partir do 1?

- Outra solução seria utilizar o comando enquanto (*while*)

```
int i = 1;
```

```
while (i <= 1000){ true
```

```
    escrever: i;
```

```
    i++;
```

```
}
```

TELA
1
2
3
...
1000

QUADRO DE MEMÓRIA									
i	1	2	3	4	5	6	7	8	9

- Como imprimir os 1000 primeiros números a partir do 1?

- Outra solução seria utilizar o comando enquanto (*while*)

```
int i = 1;  
  
while (i <= 1000){  
    escrever: i;  
    i++;  
}
```

TELA
1
2
3
...
1000

QUADRO DE MEMÓRIA									
i	1	2	3	4	5	6	7	8	9

● Como imprimir os 1000 primeiros números a partir do 1?

● Outra solução seria utilizar o comando enquanto (*while*)

```
int i = 1;  
  
while (i <= 1000){  
    escrever: i;  
    i++;  
}
```

TELA
1
2
3
...
1000

QUADRO DE MEMÓRIA				
i	1	2	3	4

- Como imprimir os 1000 primeiros números a partir do 1?

- Outra solução seria utilizar o comando enquanto (*while*)

```
int i = 1;  
  
while (i <= 1000){  
    escrever: i;  
    i++;  
}
```

TELA

1  
2  
3  
...  
9999  
1000

QUADRO DE MEMÓRIA

i	1	2	3	4	...	1000
---	---	---	---	---	-----	------

## Exercício

● Como imprimir os 1000 primeiros números a partir do 1?

● Outra solução seria utilizar o comando enquanto (*while*)

```
int i = 1;
```

```
while (i <= 1000){
```

```
    escrever: i;
```

```
    i++;
```

```
}
```

TELA
1
2
3
...
9999
11000

QUADRO DE MEMÓRIA						
i	1	2	3	4	...	1000

## Exercício

- Como imprimir os 1000 primeiros números a partir do 1?

- Outra solução seria utilizar o comando enquanto (*while*)

```
int i = 1;  
  
while (i <= 1000){  
    escrever: i;  
    i++;  
}
```

TELA

1

2

3

•••

9999

11000

QUADRO DE MEMÓRIA

i

1 2 3 4 ••• 1000



● Como imprimir os 1000 primeiros números a partir do 1?

● Outra solução seria utilizar o comando enquanto (*while*)

```
int i = 1;  
  
while (i <= 1000){  
    escrever: i;  
    i++;  
}
```

TELA

1

2

3

...

9999

10000

QUADRO DE MEMÓRIA

i

1

2

3

4

...

1000

1001


## Exercício

- Como imprimir os 1000 primeiros números a partir do 1?

- Outra solução seria utilizar o comando enquanto (*while*)

TELA
1
2
3
...
9999
11000

```
int i = 1;  
while (i <= 1000){  
    escrever: i;  
    i++;  
}
```



QUADRO DE MEMÓRIA							
i	1	2	3	4...	1000	1001	

- Faça o quadro de memória e mostre a saída na tela para o código abaixo

```
int num = 0;  
  
while (num < 4){  
    escrever: num++, " ";  
    num += 2;  
}
```

- Faça o quadro de memória e mostre a saída na tela para o código abaixo

```
int num = 0;

while (num < 4){
    escrever: num++, " ";
    num += 2;
}
```

TELA
0
3

QUADRO DE MEMÓRIA	
num	0 1 3 4 6

- Faça o quadro de memória e mostre a saída na tela para o código abaixo

```
int num = 0;
```

```
while (num < 4){  
    escrever: num++, " ";  
    num += 2;  
}
```

TELA

0

3

## QUADRO DE MEMÓRIA

num	0 1 3 4 6
-----	-----------

- Faça o quadro de memória e mostre a saída na tela para o código abaixo

```
int num = 0;
```

```
while (num < 4){ true
```

```
    escrever: num++, " ";
```

```
    num += 2;
```

```
}
```

TELA
0
3

QUADRO DE MEMÓRIA	
num	0 1 3 4 6

- Faça o quadro de memória e mostre a saída na tela para o código abaixo

```
int num = 0;  
  
while (num < 4){  
    escrever: num++, " ";  
    num += 2;  
}
```

TELA

0  
3

## QUADRO DE MEMÓRIA

QUADRO DE MEMÓRIA	
num	0 1 3 4 6

- Faça o quadro de memória e mostre a saída na tela para o código abaixo

```
int num = 0;  
  
while (num < 4){  
    escrever: num++, " ";  
    num += 2;  
}
```

TELA
0
3

QUADRO DE MEMÓRIA	
num	0 1 3 4 6



- Faça o quadro de memória e mostre a saída na tela para o código abaixo

```
int num = 0;

while (num < 4){
    escrever: num++, " ";
    num += 2;
}
```

TELA
0
3

QUADRO DE MEMÓRIA	
num	0 1 3 4 6

- Faça o quadro de memória e mostre a saída na tela para o código abaixo

```
int num = 0;
```

```
while (num < 4){ true
```

```
    escrever: num++, " ";
```

```
    num += 2;
```

```
}
```

TELA
0
3

QUADRO DE MEMÓRIA	
num	0 1 3 4 6

- Faça o quadro de memória e mostre a saída na tela para o código abaixo

```
int num = 0;  
  
while (num < 4){  
    escrever: num++, " ";  
    num += 2;  
}
```

TELA

0  
3

## QUADRO DE MEMÓRIA

num	0 1 3 4 6
-----	-----------

- Faça o quadro de memória e mostre a saída na tela para o código abaixo

```
int num = 0;  
  
while (num < 4){  
    escrever: num++, " ";  
    num += 2;  
}
```

TELA
0
3

QUADRO DE MEMÓRIA	
num	0 1 3 4 6

- Faça o quadro de memória e mostre a saída na tela para o código abaixo

```
int num = 0;  
  
while (num < 4){  
    escrever: num++, " ";  
    num += 2;  
}
```

TELA
0
3

QUADRO DE MEMÓRIA	
num	0 1 3 4 6

- Faça o quadro de memória e mostre a saída na tela para o código abaixo

```
int num = 0;  
while (num < 4){  
    escrever: num++, " ";  
    num += 2;  
}
```

**false**

TELA

0

3

## QUADRO DE MEMÓRIA

QUADRO DE MEMÓRIA	
num	0 1 3 4 6

# Algumas Observações

- O corpo de um while pode ter zero, um ou n comandos
- A lista de comandos será executada zero ou mais vezes
- O } e o { são obrigatório apenas se tivermos mais de um comando
  - Uma boa prática de programação consiste em sempre usá-los

- Faça o quadro de memória e mostre a saída na tela para o código abaixo :

```
int num = 0;  
  
while (num < 3){  
    escrever: ++num, " ";  
    num += 2;  
}
```



- Faça o quadro de memória e mostre a saída na tela para o código abaixo :

```
int num = 0;

while (num < 3){
    escrever: ++num, " ";
    num += 2;
}
```

TELA
0
3

QUADRO DE MEMÓRIA	
num	0 1 3 4 6

- Faça o quadro de memória e mostre a saída na tela para o código abaixo :

```
int num = 0;

while (num < 3){
    escrever: ++num, " ";
    num += 2;
}
```

TELA
0
3

QUADRO DE MEMÓRIA	
num	0 1 3 4 6

- Faça um programa que mostre os 10 primeiros números inteiros positivos.
- Faça um programa que leia um número inteiro  $N$  e mostre na tela os  $N$  primeiros números inteiros ímpares.
- Faça um programa que leia um número inteiro  $N$  e mostre na tela os  $N$  primeiros números da sequência 1, 5, 12, 16, 23, 27 34.

- Faça um programa que leia um número inteiro  $N$  indicando a nota máxima em uma prova  $P$ . Em seguida, leia a nota de 20 alunos (entre 0 e  $N$ ) e mostre na tela: (i) a média da turma, (ii) o número de alunos cuja nota foi menor que a média da Universidade (suponha 60%) e (iii) a quantidade de alunos com conceito A (mais de 90%).
- Faça um programa que leia um número inteiro  $n$  e mostre na tela o  $n$ -ésimo termo da sequência de Fibonacci

# Comando Faça – Enquanto

- Similar ao comando enquanto
  - A diferença entre eles é o momento em que a expressão é avaliada
- No faça-enquanto, a lista de comandos é executada e, depois, a expressão é avaliada

# Comando Faça – Enquanto

**faça**



lista de comandos

**enquanto (expressão);**

## *do-while*: Comando Faça – Enquanto

**do {**

lista de comandos

**} while (expressão);**

- Leia dois números reais e realize a divisão entre eles:



- Leia dois números reais e realize a divisão entre eles:

```
double num1, num2;  
  
escrever: Digite o primeiro número: ";  
ler num1;  
  
do {  
    escrever: Digite o segundo número: ";  
    ler num2;  
while (num == 0);  
  
escrever: Divisao: ", (num1 / num2);
```

- Leia um número inteiro e garanta que ele é um mês válido

- Leia um número inteiro e garanta que ele é um mês válido

```
int mes;  
  
do {  
    ler mes;  
while ((mes >= 1 && mes <= 12) == false);
```

- Faça um programa que leia um número inteiro e garanta que ele é um ano bissexto

- Faça um programa que leia um número inteiro e garanta que ele é um ano bissexto

```
int ano;  
  
do {  
    ler ano;  
while ((ano % 4 == 0 && ano % 100 != 0 || ano % 400 == 0) == false);
```

# Comando Para

- Similar ao comando enquanto, contudo, ele permite: pré-comandos (início) e pós-comandos (incremento)

## Comando Para

- Similar ao comando enquanto, contudo, ele permite: pré-comandos (início) e pós-comandos (incremento)

```
início;  
  
while (expressão) {  
  
    lista de comandos;  
  
    incremento;  
  
}
```

## Comando Para

- Similar ao comando enquanto, contudo, ele permite: pré-comandos (início) e pós-comandos (incremento)

```
início;  
  
while (expressão) {  
    lista de comandos;  
    incremento;  
}
```

```
int i = 1;  
  
while (i <= 1000){  
    escrever: i;  
    i++;  
}
```



**para** (início ; expressão ; incremento)

lista de comandos

**fim para**



**para** (início ; expressão ; incremento)



lista de comandos

**fim para**

**para** (**início** ; expressão ; incremento)

lista de comandos

**fim para**

**para** (**início** ; expressão ; incremento)

lista de comandos

**fim para**

The diagram illustrates a 'para' loop structure. The word 'para' is in blue. The opening parenthesis is in blue, and the closing parenthesis is in blue. The word 'início' is in red. The word 'expressão' is in black. The word 'incremento' is in black. A red arrow points from the 'início' to the 'expressão'. Another red arrow points from the 'expressão' to the 'incremento'. A third red arrow points from the 'incremento' back to the 'início', forming a loop. A blue vertical line connects the 'para' to the 'fim para'.

**para** (início ; expressão ; incremento)

lista de comandos

**fim para**

**para** (início ; expressão ; incremento)

lista de comandos

**fim para**

true or false?

**para** (início ; expressão ; incremento)

↓ true

lista de comandos

**fim para**

**para** (início ; expressão ; incremento)

lista de comandos

**fim para**



**para** (início ; expressão ; incremento)

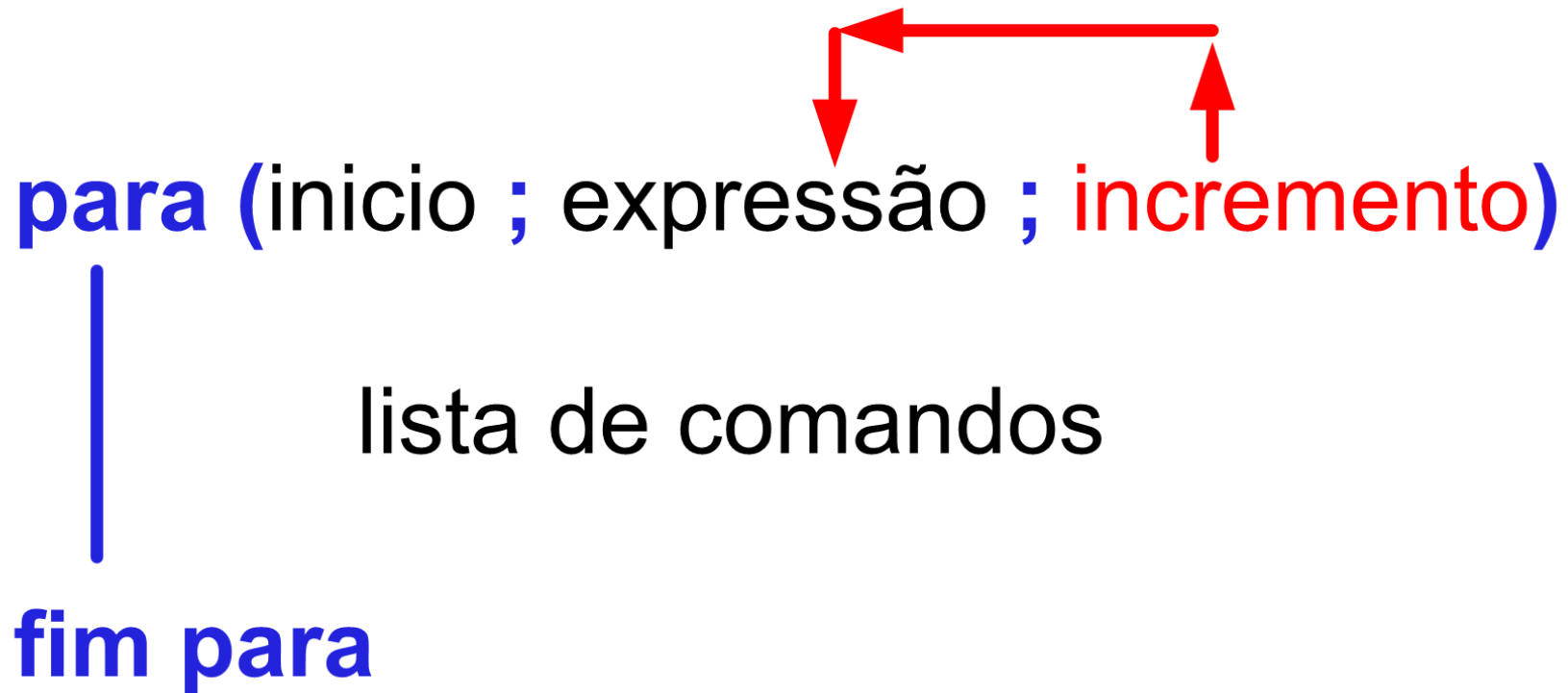
lista de comandos → ↑

**fim para**

**para** (início ; expressão ; **incremento**)

lista de comandos

**fim para**



**para** (início ; expressão ; incremento)

lista de comandos

**fim para**

The diagram illustrates the flow of a 'para' loop. A vertical blue line connects the word 'para' to 'fim para'. A red arrow points from 'expressão' down to the start of the 'lista de comandos'. Another red arrow points from the end of the 'lista de comandos' up to 'incremento'. A third red arrow points from 'incremento' back to 'expressão', forming a loop that indicates the repetition of the command list.

**para** (início ; expressão ; incremento)

lista de comandos

**fim para**

true or false?

**para** (início ; expressão ; incremento)

↓ true

lista de comandos

**fim para**

**para** (início ; expressão ; incremento)

lista de comandos

**fim para**

**para** (início ; expressão ; incremento)

lista de comandos → ↑

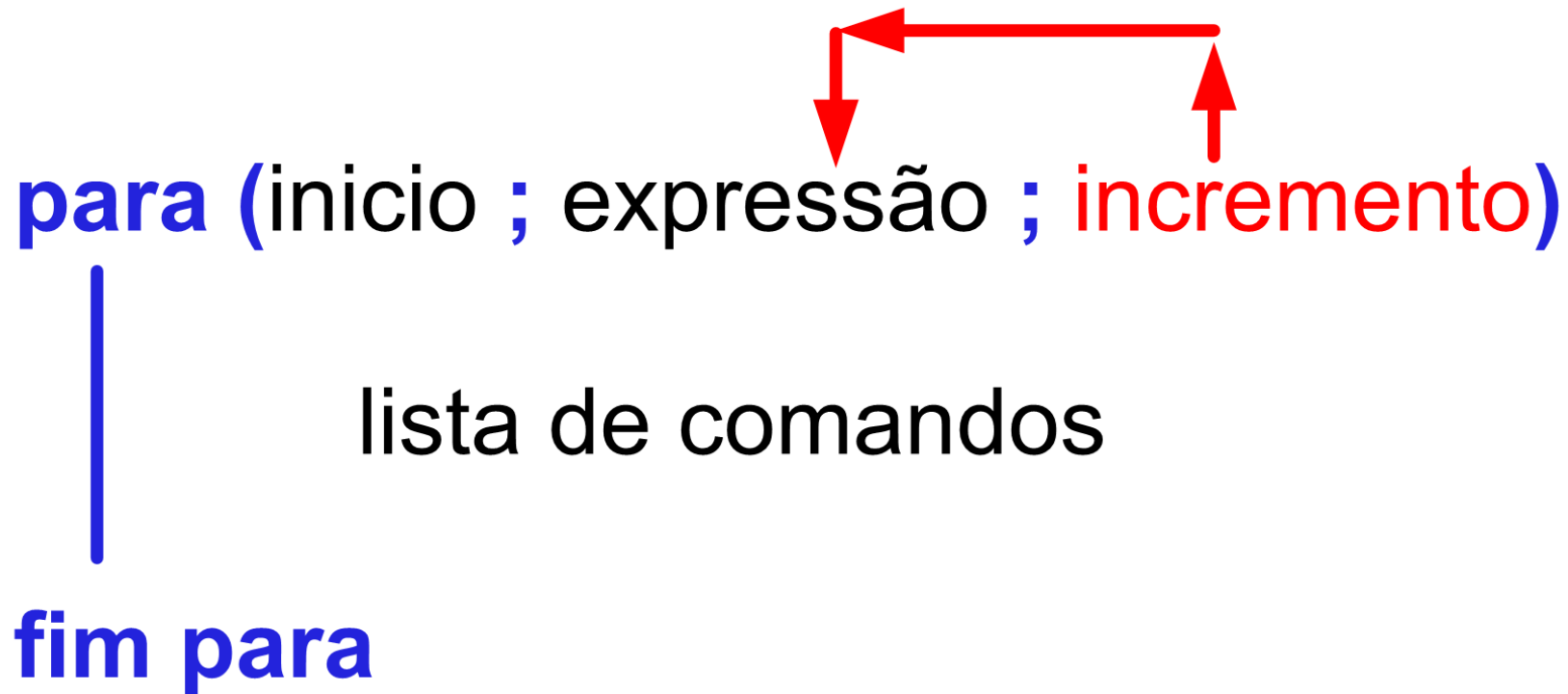
**fim para**

**para** (início ; expressão ; **incremento**)

lista de comandos

**fim para**





The diagram illustrates the syntax of a 'para' (for) loop. The text 'para (início ; expressão ; incremento)' is shown in blue, black, and red. A red arrow points from 'expressão' to 'incremento', and another red arrow points from 'incremento' back to 'expressão', forming a loop. A blue vertical line connects 'para' to 'fim para'. The text 'lista de comandos' is centered below the loop header.

```
para (início ; expressão ; incremento)  
    lista de comandos  
fim para
```

**para** (início ; expressão ; incremento)

false

lista de comandos

**fim para**

## *for*: Comando Para em C-like

```
for (início ; expressão ; incremento) {  
    lista de comandos  
}
```

- Como imprimir os 3 primeiros números a partir do 1?

```
for ( int i = 1 ; i <= 3 ; i++ ){  
    escrever: i;  
}
```

- Como imprimir os 3 primeiros números a partir do 1?

```
for ( int i = 1 ; i <= 3 ; i++ ){  
    escrever: i;  
}
```

TELA
1
2
3
...
1000

QUADRO DE MEMÓRIA				
i	1	2	3	4

- Como imprimir os 3 primeiros números a partir do 1?

```
for (int i = 1 ; i <= 3 ; i++ ){  
    escrever: i;  
}
```

TELA
1
2
3
...
1000

QUADRO DE MEMÓRIA				
i	1	2	3	4

- Como imprimir os 3 primeiros números a partir do 1?

```
for ( int i = 1 ; true i <= 3 ; i++ ){  
    escrever: i;  
}
```

TELA
1
2
3
...
1000

QUADRO DE MEMÓRIA				
i	1	2	3	4

- Como imprimir os 3 primeiros números a partir do 1?

```
for ( int i = 1 ; i <= 3 ; i++ ){  
    escrever: i;  
}
```

TELA
1
2
3
...
1000

QUADRO DE MEMÓRIA				
i	1	2	3	4



- Como imprimir os 3 primeiros números a partir do 1?

```
for ( int i = 1 ; i <= 3 ; i++ ){  
    escrever: i;  
}
```

TELA
1
2
3
...
1000

QUADRO DE MEMÓRIA				
i	1	2	3	4

- Como imprimir os 3 primeiros números a partir do 1?

```
for ( int i = 1 ; true i <= 3 ; i++ ){  
    escrever: i;  
}
```

TELA
1
2
3
...
1000

QUADRO DE MEMÓRIA				
i	1	2	3	4

- Como imprimir os 3 primeiros números a partir do 1?

```
for ( int i = 1 ; i <= 3 ; i++ ){  
    escrever: i;  
}
```

TELA
1
2
3
...
1000

QUADRO DE MEMÓRIA				
i	1	2	3	4

- Como imprimir os 3 primeiros números a partir do 1?

```
for ( int i = 1 ; i <= 3 ; i++) {  
    escrever: i;  
}
```

TELA
1
2
3
...
1000

QUADRO DE MEMÓRIA				
i	1	2	3	4

- Como imprimir os 3 primeiros números a partir do 1?

```
for ( int i = 1 ; i <= 3 ; i++ ){  
    escrever: i;  
}
```

TELA
1
2
3
...
1000

QUADRO DE MEMÓRIA				
i	1	2	3	4

- Como imprimir os 3 primeiros números a partir do 1?

```
for ( int i = 1 ; i <= 3 ; i++ ){  
    escrever: i;  
}
```

TELA
1
2
3
...
1000

QUADRO DE MEMÓRIA				
i	1	2	3	4

- Como imprimir os 3 primeiros números a partir do 1?

```
for ( int i = 1 ; i <= 3 ; i++) {  
    escrever: i;  
}
```


TELA
1
2
3
...
1000

QUADRO DE MEMÓRIA				
i	1	2	3	4

- Como imprimir os 3 primeiros números a partir do 1?

```
for ( int i = 1 ; i <= 3 ; i++ ){  
    escrever: i;  
}
```

**false**



## QUADRO DE MEMÓRIA

i	1	2	3	4
---	---	---	---	---

## TELA

1

2

3

...

1000



- Faça o quadro de memória e mostre a saída na tela para o código abaixo :

```
for ( int i = 0 ; i < 10 ; i += 2 ){  
    escrever: i++, " ";  
}
```

TELA
1
2
3
...
1000

QUADRO DE MEMÓRIA				
i	1	2	3	4

- Faça um programa para ler um número inteiro  $n$  e mostrar na tela os  $n$  primeiros números inteiros e positivos

```
int n;  
  
ler n;  
  
for (int i = 1; i < n; i += 1){  
    escrever: i;  
}
```

- Faça um programa para imprimir os quatro primeiros múltiplos de cinco

```
for ( int i = 0, valor = 5 ; i < 4 ; i += 1, valor += 5 ){  
    escrever: valor, " ";  
}
```

- Faça um programa para ler dois números inteiros e multiplicá-los sem utilizar a operação de multiplicação

**Algoritmo**

```
int n1, n2, i, soma;
escrever: "Digite o multiplicando: ";
ler n1;
escrever: "Digite o multiplicador: ";
ler n2;
for(int i = 1, soma = 0; i <= n2; i++, soma = soma + n1);
escrever: "Produto: ", soma;
```

**Fim Algoritmo**

- Faça um programa para ler um número inteiro  $n$  e mostrar na tela os  $n$  primeiros termos da sequência 1, 5, 12, 16, 23, 27, 34, ... que sejam divisíveis por dois and não divisíveis por três

- Faça um programa para ler um número  $n$  do teclado e calcular o  $n$ -ésimo termo da sequência de Fibonacci (1, 1, 2, 3, 5, 8, 13, 21, ...) utilizando: (a) o comando **for**; (b) o comando **do-while**.

● **Desafio:** Sabendo que os alunos de uma turma com  $N_A$  fizeram  $N_P$  provas, faça um programa para ler cada uma das  $N_P$  provas feitas por cada um dos  $N_A$  alunos e mostrar na tela (i) a média de cada aluno, (ii) a média da turma e o (iii) percentual dos alunos cuja média foi maior ou igual a 80%

# LOONEY TUNES



*"That's all Folks!"*