

Aula anterior

Programação defensiva e
Tratamento de Exceção

AEDs II – Recursividade

Prof. Diego Silva Caldeira Rocha

Objetivos

- Introdução a Recursividade
- Fundamentos da Recursividade
 - Implementação de Recursividade
 - As três regras de Recursão
 - Quando Não usar Recursividade

- Definição: Um método é dito recursivo quando ele faz chamada a si próprio
- Um método recursivo normalmente apresenta duas características básicas:
 - Chamada recursiva
 - Condição de parada



Foto recursiva



Imagem recursiva



Pensamento recursivo

- Identifique as chamadas recursivas e condições de parada

```
int fat (int n){  
    int resp;  
    if (n == 1){  
        resp = 1;  
    } else {  
        resp = n * fat (n - 1);  
    }  
    return resp;  
}
```

Chamadas recursivas

```
int fib (int n){  
    int resp;  
    if (n == 0 || n == 1){  
        resp = 1;  
    } else {  
        resp = fib (n - 1) + fib(n - 2);  
    }  
    return resp;  
}
```

- Identifique as chamadas recursivas e condições de parada

```
int fat (int n){  
    int resp;  
    if (n == 1){  
        resp = 1;  
    } else {  
        resp = n * fat (n - 1);  
    }  
    return resp;  
}
```

A cada chamada recursiva,
o n se aproxima do último valor

Condições de parada

```
int fib (int n){  
    int resp;  
    if (n == 0 || n == 1){  
        resp = 1;  
    } else {  
        resp = fib (n - 1) + fib(n - 2);  
    }  
    return resp;  
}
```

- Identifique as chamadas recursivas e condições de parada

```
int fat (int n){  
    int resp;  
    if (n == 1){  
        resp = 1;  
    } else {  
        resp = n * fat (n - 1);  
    }  
    return resp;  
}
```

Algoritmo	Código em Java
Print(i)	System.out.println
void main (...)	public static void main(String args[])

```
int fib (int n){  
    int resp;  
    if (n == 0 || n == 1){  
        resp = 1;  
    } else {  
        resp = fib (n - 1) + fib(n - 2);  
    }  
    return resp;  
}
```


Exemplo de Método Iterativo

- O que o programa iterativo abaixo mostra na tela?

```
void primeiro(){  
    print("1o – inicio");  
    segundo();  
    print("1o – fim");  
}  
  
void segundo(){  
    print("2o – inicio e fim");  
}  
  
void main (){  
    print("main – inicio");  
    primeiro();  
    print("main – fim");  
}
```

Exemplo de Método Iterativo

- O que o programa iterativo abaixo mostra na tela?

```
void primeiro(){  
    print("1o – início");  
    segundo();  
    print("1o – fim");  
}  
  
void segundo(){  
    print("2o – início e fim");  
}  
  
void main (){  
    print("main – início");  
    primeiro();  
    print("main – fim");  
}
```

TELA

Exemplo de Método Iterativo

- O que o programa iterativo abaixo mostra na tela?

```
void primeiro(){  
    print("1o – início");  
    segundo();  
    print("1o – fim");  
}  
  
void segundo(){  
    print("2o – início e fim");  
}  
  
void main (){  
    print("main – início");  
    primeiro();  
    print("main – fim");  
}
```

TELA

main – início

Exemplo de Método Iterativo

- O que o programa iterativo abaixo mostra na tela?

```
void primeiro(){  
    print("1o – início");  
    segundo();  
    print("1o – fim");  
}  
  
void segundo(){  
    print("2o – início e fim");  
}  
  
void main (){  
    print("main – início");  
    primeiro();  
    print("main – fim");  
}
```

TELA

main – início

Exemplo de Método Iterativo

- O que o programa iterativo abaixo mostra na tela?

```
void primeiro(){  
    print("1o – início");  
    segundo();  
    print("1o – fim");  
}  
  
void segundo(){  
    print("2o – início e fim");  
}  
  
void main (){  
    print("main – início");  
    primeiro();  
    print("main – fim");  
}
```

TELA

main – início

Exemplo de Método Iterativo

- O que o programa iterativo abaixo mostra na tela?

```
void primeiro(){  
    print("1o – início");  
    segundo();  
    print("1o – fim");  
}  
  
void segundo(){  
    print("2o – início e fim");  
}  
  
void main (){  
    print("main – início");  
    primeiro();  
    print("main – fim");  
}
```

TELA

main – início

1º – início

Exemplo de Método Iterativo

- O que o programa iterativo abaixo mostra na tela?

```
void primeiro(){  
    print("1o – início");  
    segundo();  
    print("1o – fim");  
}  
  
void segundo(){  
    print("2o – início e fim");  
}  
  
void main (){  
    print("main – início");  
    primeiro();  
    print("main – fim");  
}
```

TELA

main – início

1º – início

Exemplo de Método Iterativo

- O que o programa iterativo abaixo mostra na tela?

```
void primeiro(){  
    print("1o – início");  
    segundo();  
    print("1o – fim");  
}
```

```
void segundo(){  
    print("2o – início e fim");  
}
```

```
void main (){  
    print("main – início");  
    primeiro();  
    print("main – fim");  
}
```

TELA

main – início

1º – início

Exemplo de Método Iterativo

- O que o programa iterativo abaixo mostra na tela?

```
void primeiro(){  
    print("1o – início");  
    segundo();  
    print("1o – fim");  
}  
  
void segundo(){  
    print("2o – início e fim");  
}  
  
void main (){  
    print("main – início");  
    primeiro();  
    print("main – fim");  
}
```

TELA

main – início

1º – início

2º – início e fim

Exemplo de Método Iterativo

- O que o programa iterativo abaixo mostra na tela?

```
void primeiro(){  
    print("1o – início");  
    segundo();  
    print("1o – fim");  
}  
  
void segundo(){  
    print("2o – início e fim");  
}  
  
void main (){  
    print("main – início");  
    primeiro();  
    print("main – fim");  
}
```

TELA

main – início

1º – início

2º – início e fim

Exemplo de Método Iterativo

- O que o programa iterativo abaixo mostra na tela?

```
void primeiro(){  
    print("1o – início");  
    segundo();  
    print("1o – fim");  
}  
  
void segundo(){  
    print("2o – início e fim");  
}  
  
void main (){  
    print("main – início");  
    primeiro();  
    print("main – fim");  
}
```

TELA

main – início
1º – início
2º – início e fim
1º – fim

Exemplo de Método Iterativo

- O que o programa iterativo abaixo mostra na tela?

```
void primeiro(){  
    print("1o – início");  
    segundo();  
    print("1o – fim");  
}  
  
void segundo(){  
    print("2o – início e fim");  
}  
  
void main (){  
    print("main – início");  
    primeiro();  
    print("main – fim");  
}
```

TELA

main – início

1º – início

2º – início e fim

1º – fim

Exemplo de Método Iterativo

- O que o programa iterativo abaixo mostra na tela?

```
void primeiro(){  
    print("1o – início");  
    segundo();  
    print("1o – fim");  
}  
  
void segundo(){  
    print("2o – início e fim");  
}  
  
void main (){  
    print("main – início");  
    primeiro();  
    print("main – fim");  
}
```

TELA

main – início
1º – início
2º – início e fim
1º – fim
main – fim

- Por que o código abaixo imprime 2, 1, 0, 0, 1 e 2?

```
void printRecursivo(){  
    printRecursivo(2);  
}  
  
void printRecursivo(int i){  
    print(i);  
    if (i > 0){  
        printRecursivo(i - 1);  
    }  
    print(i);  
}
```

- Por que o código abaixo imprime 2, 1, 0, 0, 1 e 2?

```
void printRecursivo(){  
    printRecursivo(2);  
}  
  
void printRecursivo(int i){  
    print(i);  
    if (i > 0){  
        printRecursivo(i - 1);  
    }  
    print(i);  
}
```

Temos como se cada chamada recursiva fosse um método diferente!!!

Exercício - Reavaliando

- Por que o código abaixo imprime 2, 1, 0, 0, 1 e 2?

```
void printRecursivo(){  
    printRecursivo(2);  
}
```

```
void printRecursivo(int i){ //i(2)  
    print(i);  
    if (i > 0){  
        printRecursivo(i - 1);  
    }  
    print(i);  
}
```

```
void printRecursivo(int i){ //i(1)  
    print(i);  
    if (i > 0){  
        printRecursivo(i - 1);  
    }  
    print(i);  
}
```

```
void printRecursivo(int i){ //i(0)  
    print(i);  
    if (i > 0){  
        printRecursivo(i - 1);  
    }  
    print(i);  
}
```


Exercício - Reavaliando

- Por que o código abaixo imprime 2, 1, 0, 0, 1 e 2?

```
(1) void printRecursivo(){  
(2)     printRecursivo(2);  
}
```

```
(3) void printRecursivo(int i){ // i (2)  
(4)     print(i);  
(5)     if (i > 0){  
(6)         printRecursivo(i - 1);  
        }  
        print(i);  
}
```

```
(7) void printRecursivo(int i){ // i (1)  
(8)     print(i);  
(9)     if (i > 0){  
(10)        printRecursivo(i - 1);  
        }  
        print(i);  
}
```

```
(11) void printRecursivo(int i){ // i (0)  
(12)     print(i);  
(13)     if (i > 0){  
        printRecursivo(i - 1);  
        }  
(14)     print(i);  
}
```

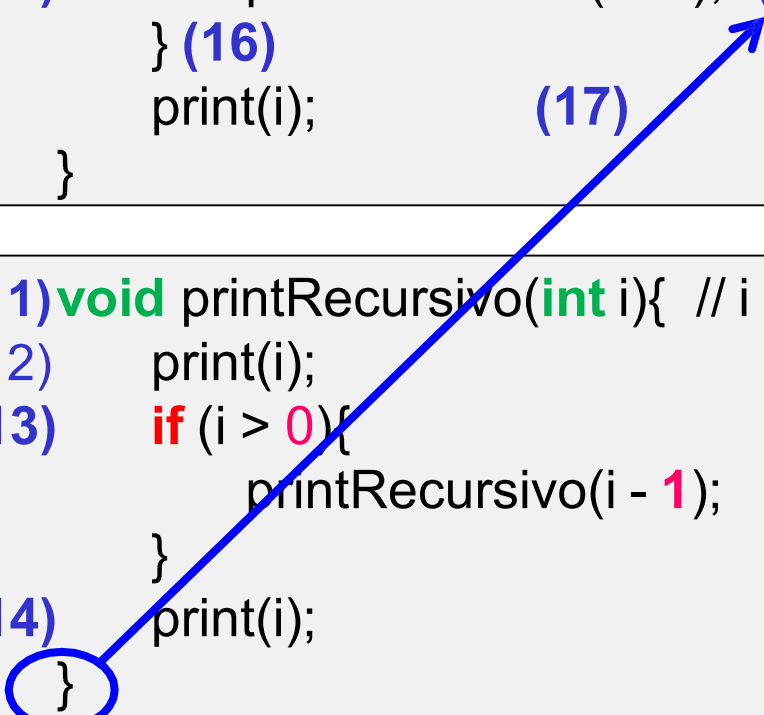
Exercício - Reavaliando

- Por que o código abaixo imprime 2, 1, 0, 0, 1 e 2?

```
(1) void printRecursivo(){  
(2)     printRecursivo(2);  
}
```

```
(3) void printRecursivo(int i){ // i (2)  
(4)     print(i);  
(5)     if (i > 0){  
(6)         printRecursivo(i - 1);  
        }  
        print(i);  
}
```

```
(7) void printRecursivo(int i){ // i (1)  
(8)     print(i);  
(9)     if (i > 0){  
(10)        printRecursivo(i - 1); (15)  
        } (16)  
        print(i); (17)  
}
```



```
(11) void printRecursivo(int i){ // i (0)  
(12)     print(i);  
(13)     if (i > 0){  
        printRecursivo(i - 1);  
    }  
(14)     print(i);  
    }
```

Exercício - Reavaliando

- Por que o código abaixo imprime 2, 1, 0, 0, 1 e 2?

```
(1) void printRecursivo(){  
(2)     printRecursivo(2);  
}
```

```
(3) void printRecursivo(int i){ // i (2)  
(4)     print(i);  
(5)     if (i > 0){  
(6)         printRecursivo(i - 1); (18)  
        } (19)  
        print(i); (20)  
}
```

```
(7) void printRecursivo(int i){ // i (1)  
(8)     print(i);  
(9)     if (i > 0){  
(10)        printRecursivo(i - 1); (15)  
        } (16)  
        print(i); (17)  
}
```

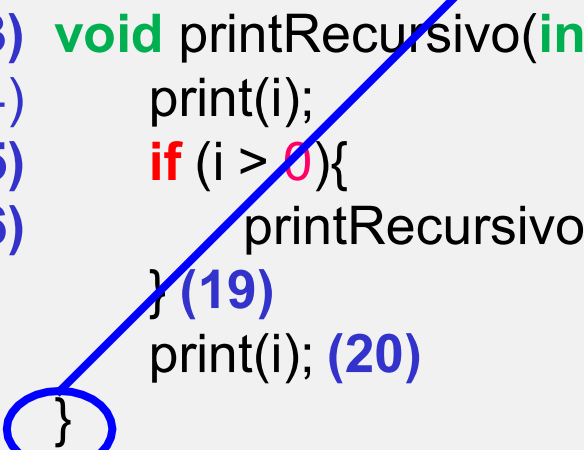
```
(11) void printRecursivo(int i){ // i (0)  
(12)     print(i);  
(13)     if (i > 0){  
        printRecursivo(i - 1);  
    }  
(14)     print(i);  
}
```

Exercício - Reavaliando

- Por que o código abaixo imprime 2, 1, 0, 0, 1 e 2?

```
(1) void printRecursivo(){
(2)     printRecursivo(2); (21)
}
```

```
(3) void printRecursivo(int i){ // i (2)
(4)     print(i);
(5)     if (i > 0){
(6)         printRecursivo(i - 1); (18)
            } (19)
        print(i); (20)
    }
```



```
(7) void printRecursivo(int i){ // i (1)
(8)     print(i);
(9)     if (i > 0){
(10)         printRecursivo(i - 1); (15)
            } (16)
        print(i); (17)
    }
```

```
(11) void printRecursivo(int i){ // i (0)
(12)     print(i);
(13)     if (i > 0){
            printRecursivo(i - 1);
        }
(14)     print(i);
    }
```

Exercício - Reavaliando

- Por que o código abaixo imprime 2, 1, 0, 0, 1 e 2?

```
(1) void printRecursivo(){
(2)     printRecursivo(2); (21)
    }
```

```
(3) void printRecursivo(int i){ // i (2)
(4)     print(i);
(5)     if (i > 0){
(6)         printRecursivo(i - 1); (18)
    } (19)
    print(i); (20)
}
```

```
(7) void printRecursivo(int i){ // i (1)
(8)     print(i);
(9)     if (i > 0){
(10)        printRecursivo(i - 1); (15)
    } (16)
    print(i); (17)
}
```

```
(11) void printRecursivo(int i){ // i (0)
(12)     print(i);
(13)     if (i > 0){
        printRecursivo(i - 1);
    }
(14)     print(i);
}
```

Exercício - Reavaliando

- Por que o código abaixo imprime 2, 1, 0, 0, 1 e 2?

```
void printRecursivo(){  
    printRecursivo(2);  
}
```

```
void printRecursivo(int i){ // i (2)  
    print(i);  
    if (i > 0){  
        printRecursivo(i - 1);  
    }  
    print(i);  
}
```

```
void printRecursivo(int i){ // i (1)  
    print(i);  
    if (i > 0){  
        printRecursivo(i - 1);  
    }  
    print(i);  
}
```

```
void printRecursivo(int i){ // i (0)  
    print(i);  
    if (i > 0){  
        printRecursivo(i - 1);  
    }  
    print(i);  
}
```

Exercício - Reavaliando

- Por que o código abaixo imprime 2, 1, 0, 0, 1 e 2?

```
void printRecursivo(){  
    printRecursivo(2);  
}
```

```
void printRecursivo(int i){ // i (2)  
    print(i);  
    if (i > 0){  
        printRecursivo(i - 1);  
    }  
    print(i);  
}
```

```
void printRecursivo(int i){ // i (1)  
    print(i);  
    if (i > 0){  
        printRecursivo(i - 1);  
    }  
    print(i);  
}
```

```
void printRecursivo(int i){ // i (0)  
    print(i);  
    if (i > 0){  
        printRecursivo(i - 1);  
    }  
    print(i);  
}
```

Exercício - Reavaliando

- Por que o código abaixo imprime 2, 1, 0, 0, 1 e 2?

```
void printRecursivo(){  
    printRecursivo(2);  
}
```



```
void printRecursivo(int i){ // i (2)  
    print(i);  
    if (i > 0){  
        printRecursivo(i - 1);  
    }  
    print(i);  
}
```

```
void printRecursivo(int i){ // i (1)  
    print(i);  
    if (i > 0){  
        printRecursivo(i - 1);  
    }  
    print(i);  
}
```

```
void printRecursivo(int i){ // i (0)  
    print(i);  
    if (i > 0){  
        printRecursivo(i - 1);  
    }  
    print(i);  
}
```


Exercício - Reavaliando

- Por que o código abaixo imprime 2, 1, 0, 0, 1 e 2?

```
void printRecursivo(){  
    printRecursivo(2);  
}
```



```
void printRecursivo(int i){ // i (2)  
    print(i);  
    if (i > 0){  
        printRecursivo(i - 1);  
    }  
    print(i);  
}
```

```
void printRecursivo(int i){ // i (1)  
    print(i);  
    if (i > 0){  
        printRecursivo(i - 1);  
    }  
    print(i);  
}
```

```
void printRecursivo(int i){ // i (0)  
    print(i);  
    if (i > 0){  
        printRecursivo(i - 1);  
    }  
    print(i);  
}
```

Exercício - Reavaliando

- Por que o código abaixo imprime 2, 1, 0, 0, 1 e 2?

```
void printRecursivo(){  
    printRecursivo(2);  
}
```



```
void printRecursivo(int i){ // i (2)  
    print(i);  
    if (i > 0){  
        printRecursivo(i - 1);  
    }  
    print(i);  
}
```

true

```
void printRecursivo(int i){ // i (1)  
    print(i);  
    if (i > 0){  
        printRecursivo(i - 1);  
    }  
    print(i);  
}
```

```
void printRecursivo(int i){ // i (0)  
    print(i);  
    if (i > 0){  
        printRecursivo(i - 1);  
    }  
    print(i);  
}
```

Exercício - Reavaliando

- Por que o código abaixo imprime 2, 1, 0, 0, 1 e 2?

```
void printRecursivo(){
    printRecursivo(2);
}
```



```
void printRecursivo(int i){ // i (2)
    print(i);
    if (i > 0){
        (a)
    }
    print(i);
}
```

Vamos para o print do um,
contudo, depois, voltaremos para (a)

```
void printRecursivo(int i){ // i (1)
    print(i);
    if (i > 0){
        printRecursivo(i - 1);
    }
    print(i);
}
```

```
void printRecursivo(int i){ // i (0)
    print(i);
    if (i > 0){
        printRecursivo(i - 1);
    }
    print(i);
}
```

Exercício - Reavaliando

- Por que o código abaixo imprime 2, 1, 0, 0, 1 e 2?

```
void printRecursivo(){  
    printRecursivo(2);  
}
```

```
void printRecursivo(int i){ // i (2)  
    print(i);  
    if (i > 0){  
        printRecursivo(i - 1);  
    }  
    print(i);  
}
```

```
void printRecursivo(int i){ // i (1)  
    print(i);  
    if (i > 0){  
        printRecursivo(i - 1);  
    }  
    print(i);  
}
```

```
void printRecursivo(int i){ // i (0)  
    print(i);  
    if (i > 0){  
        printRecursivo(i - 1);  
    }  
    print(i);  
}
```

Exercício - Reavaliando

- Por que o código abaixo imprime 2, 1, 0, 0, 1 e 2?

```
void printRecursivo(){  
    printRecursivo(2);  
}
```

```
void printRecursivo(int i){ // i (2)  
    print(i);  
    if (i > 0){  
        printRecursivo(i - 1);  
    }  
    print(i);  
}
```

```
void printRecursivo(int i){ // i (1)  
    print(i);  
    if (i > 0){  
        printRecursivo(i - 1);  
    }  
    print(i);  
}
```

```
void printRecursivo(int i){ // i (0)  
    print(i);  
    if (i > 0){  
        printRecursivo(i - 1);  
    }  
    print(i);  
}
```

Exercício - Reavaliando

- Por que o código abaixo imprime 2, 1, 0, 0, 1 e 2?

```
void printRecursivo(){  
    printRecursivo(2);  
}
```

```
void printRecursivo(int i){ // i (2)  
    print(i);  
    if (i > 0){  
        printRecursivo(i - 1);  
    }  
    print(i);  
}
```

```
void printRecursivo(int i){ // i (1)  
    print(i);  
    if (i > 0){  
        printRecursivo(i - 1);  
    }  
    print(i);  
}
```

true

```
void printRecursivo(int i){ // i (0)  
    print(i);  
    if (i > 0){  
        printRecursivo(i - 1);  
    }  
    print(i);  
}
```

Exercício - Reavaliando

- Por que o código abaixo imprime 2, 1, 0, 0, 1 e 2?

```
void printRecursivo(){
    printRecursivo(2);
}
```

```
void printRecursivo(int i){ // i (2)
    print(i);
    if (i > 0){
        printRecursivo(i - 1);
    }
    print(i);
}
```

Vamos para o print do zero,
contudo, depois, voltaremos para (b)

```
void printRecursivo(int i){ // i (1)
    print(i);
    if (i > 0){
        (b)
    }
    print(i);
}
```

```
void printRecursivo(int i){ // i (0)
    print(i);
    if (i > 0){
        printRecursivo(i - 1);
    }
    print(i);
}
```

Exercício - Reavaliando

- Por que o código abaixo imprime 2, 1, 0, 0, 1 e 2?

```
void printRecursivo(){  
    printRecursivo(2);  
}
```

```
void printRecursivo(int i){ // i (2)  
    print(i);  
    if (i > 0){  
        printRecursivo(i - 1);  
    }  
    print(i);  
}
```

```
void printRecursivo(int i){ // i (1)  
    print(i);  
    if (i > 0){  
        printRecursivo(i - 1);  
    }  
    print(i);  
}
```

```
void printRecursivo(int i){ // i (0)  
    print(i);  
    if (i > 0){  
        printRecursivo(i - 1);  
    }  
    print(i);  
}
```


Exercício - Reavaliando

- Por que o código abaixo imprime **2**, **1**, **0**, 0, 1 e 2?

```
void printRecursivo(){  
    printRecursivo(2);  
}
```

```
void printRecursivo(int i){ // i (2)  
    print(i);  
    if (i > 0){  
        printRecursivo(i - 1);  
    }  
    print(i);  
}
```

```
void printRecursivo(int i){ // i (1)  
    print(i);  
    if (i > 0){  
        printRecursivo(i - 1);  
    }  
    print(i);  
}
```

```
void printRecursivo(int i){ // i (0)  
    print(i);  
    if (i > 0){  
        printRecursivo(i - 1);  
    }  
    print(i);  
}
```

Exercício - Reavaliando

- Por que o código abaixo imprime **2**, **1**, **0**, 0, 1 e 2?

```
void printRecursivo(){
    printRecursivo(2);
}
```

```
void printRecursivo(int i){ // i (2)
    print(i);
    if (i > 0){
        printRecursivo(i - 1);
    }
    print(i);
}
```

```
void printRecursivo(int i){ // i (1)
    print(i);
    if (i > 0){
        printRecursivo(i - 1);
    }
    print(i);
}
```

```
void printRecursivo(int i){ // i (0)
    print(i);
    if (i > 0){
        printRecursivo(i - 1);
    }
    print(i);
}
```

false

Exercício - Reavaliando

- Por que o código abaixo imprime **2**, **1**, **0**, **0**, 1 e 2?

```
void printRecursivo(){  
    printRecursivo(2);  
}
```

```
void printRecursivo(int i){ // i (2)  
    print(i);  
    if (i > 0){  
        printRecursivo(i - 1);  
    }  
    print(i);  
}
```

```
void printRecursivo(int i){ // i (1)  
    print(i);  
    if (i > 0){  
        printRecursivo(i - 1);  
    }  
    print(i);  
}
```

```
void printRecursivo(int i){ // i (0)  
    print(i);  
    if (i > 0){  
        printRecursivo(i - 1);  
    }  
    print(i);  
}
```

Exercício - Reavaliando

- Por que o código abaixo imprime **2**, **1**, **0**, **0**, 1 e 2?

```
void printRecursivo(){  
    printRecursivo(2);  
}
```

```
void printRecursivo(int i){ // i (2)  
    print(i);  
    if (i > 0){  
        printRecursivo(i - 1);  
    }  
    print(i);  
}
```

Voltando para (b)

```
void printRecursivo(int i){ // i (1)  
    print(i);  
    if (i > 0){  
        printRecursivo(i - 1); (b)  
    }  
    print(i);  
}
```

```
void printRecursivo(int i){ // i (0)  
    print(i);  
    if (i > 0){  
        printRecursivo(i - 1);  
    }  
    print(i);  
}
```

Exercício - Reavaliando

- Por que o código abaixo imprime **2**, **1**, **0**, **0**, **1** e **2**?

```
void printRecursivo(){  
    printRecursivo(2);  
}
```

```
void printRecursivo(int i){ // i (2)  
    print(i);  
    if (i > 0){  
        printRecursivo(i - 1);  
    }  
    print(i);  
}
```

Voltando para (b)

```
void printRecursivo(int i){ // i (1)  
    print(i);  
    if (i > 0){  
        (b)  
    }  
    print(i);  
}
```

Exercício - Reavaliando

- Por que o código abaixo imprime 2, 1, 0, 0, 1 e 2?

```
void printRecursivo(){  
    printRecursivo(2);  
}
```

```
void printRecursivo(int i){ // i (2)  
    print(i);  
    if (i > 0){  
        printRecursivo(i - 1);  
    }  
    print(i);  
}
```

```
void printRecursivo(int i){ // i (1)  
    print(i);  
    if (i > 0){  
        printRecursivo(i - 1);  
    }  
    print(i);  
}
```

Exercício - Reavaliando

- Por que o código abaixo imprime **2**, **1**, **0**, **0**, **1** e **2**?

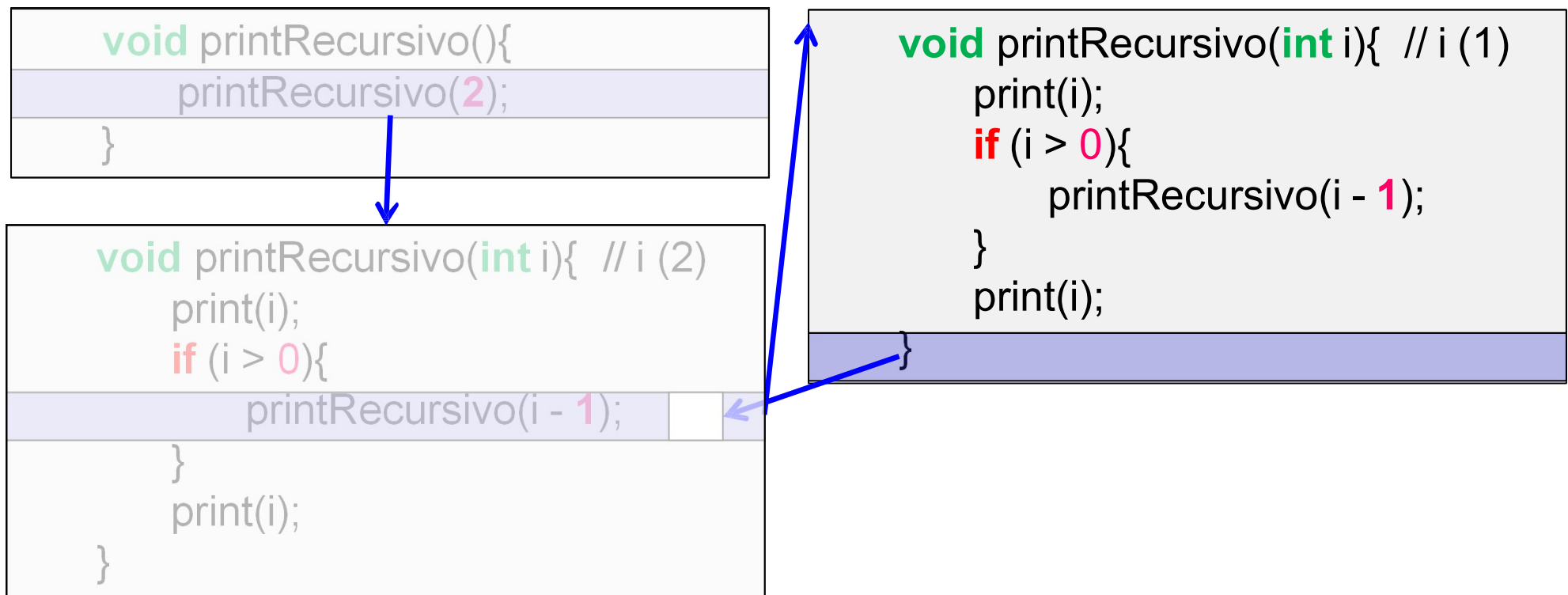
```
void printRecursivo(){  
    printRecursivo(2);  
}
```

```
void printRecursivo(int i){ // i (2)  
    print(i);  
    if (i > 0){  
        printRecursivo(i - 1);  
    }  
    print(i);  
}
```

```
void printRecursivo(int i){ // i (1)  
    print(i);  
    if (i > 0){  
        printRecursivo(i - 1);  
    }  
    print(i);  
}
```

Exercício - Reavaliando

- Por que o código abaixo imprime **2**, **1**, **0**, **0**, **1** e **2**?



Voltando para (a)

Exercício - Reavaliando

- Por que o código abaixo imprime **2**, **1**, **0**, **0**, **1** e **2**?

```
void printRecursivo(){  
    printRecursivo(2);  
}
```

```
void printRecursivo(int i){ // i (2)  
    print(i);  
    if (i > 0){  
        (a)  
    }  
    print(i);  
}
```

Voltando para (a)

Exercício - Reavaliando

- Por que o código abaixo imprime **2**, **1**, **0**, **0**, **1** e **2**?

```
void printRecursivo(){  
    printRecursivo(2);  
}
```



```
void printRecursivo(int i){ // i (2)  
    print(i);  
    if (i > 0){  
        printRecursivo(i - 1);  
    }  
    print(i);  
}
```

Exercício - Reavaliando

- Por que o código abaixo imprime **2**, **1**, **0**, **0**, **1** e **2**?

```
void printRecursivo(){  
    printRecursivo(2);  
}
```



```
void printRecursivo(int i){ // i (2)  
    print(i);  
    if (i > 0){  
        printRecursivo(i - 1);  
    }  
    print(i);  
}
```

Exercício - Reavaliando

- Por que o código abaixo imprime **2**, **1**, **0**, **0**, **1** e **2**?

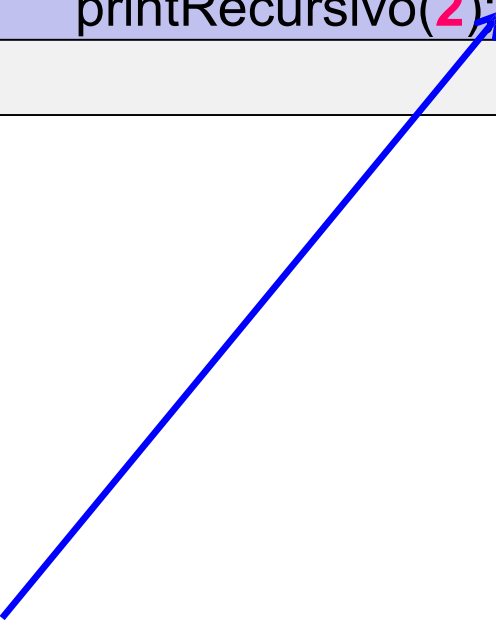
```
void printRecursivo(){  
    printRecursivo(2);  
}  
  
void printRecursivo(int i){ // i (2)  
    print(i);  
    if (i > 0){  
        printRecursivo(i - 1);  
    }  
    print(i);  
}
```

Voltando para **(primeiro)**

Exercício - Reavaliando

- Por que o código abaixo imprime **2**, **1**, **0**, **0**, **1** e **2**?

```
void printRecursivo(){  
    printRecursivo(2);  
}
```



Exercício - Reavaliando

- Por que o código abaixo imprime **2**, **1**, **0**, **0**, **1** e **2**?

```
void printRecursivo(){  
    printRecursivo(2);  
}
```

Exemplo de Método Iterativo vs Recursivo

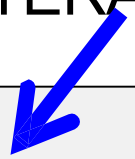
- Faça métodos ITERATIVO e RECURSIVO para mostrar os números 0 à 3

```
void mostrar (){  
    for (int i = 0; i < 4; i = i + 1) {  
        print(i);  
    }  
}
```


```
void mostrar () {  
    mostrar (0);  
}  
void mostrar (int i){  
    if (i < 4) {  
        print(i);  
        mostrar (i + 1);  
    }  
}
```

Exemplo de Método Iterativo vs Recursivo

- Faça métodos ITERATIVO e RECURSIVO para mostrar os números 0 à 3



```
void mostrar () {  
    for (int i = 0; i < 4; i = i + 1) {  
        print(i);  
    }  
}
```



```
void mostrar () {  
    mostrar (0);  
}  
void mostrar (int i) {  
    if (i < 4) {  
        print(i);  
        mostrar (i + 1);  
    }  
}
```


Exemplo de Método Iterativo vs Recursivo

- Faça métodos ITERATIVO e RECURSIVO para mostrar os números 0 à 3

```
void mostrar () {  
    for (int i = 0; i < 4; i = i + 1) {  
        print(i);  
    }  
}
```

Tela

```
void mostrar () {  
    mostrar (0);  
}  
void mostrar (int i) {  
    if (i < 4) {  
        print(i);  
        mostrar (i + 1);  
    }  
}
```

Exemplo de Método Iterativo vs Recursivo

- Faça métodos ITERATIVO e RECURSIVO para mostrar os números 0 à 3

```
void mostrar () {  
    for (int i = 0; i < 4; i = i + 1) {  
        print(i);  
    }  
}
```

Tela

```
void mostrar () {  
    mostrar (0);  
}  
void mostrar (int i) {  
    if (i < 4) {  
        print(i);  
        mostrar (i + 1);  
    }  
}
```

i	0
---	---

Exemplo de Método Iterativo vs Recursivo

- Faça métodos ITERATIVO e RECURSIVO para mostrar os números 0 à 3

```
void mostrar () {  
    for (int i = 0; i < 4; i = i + 1) {  
        print(i);  
    }  
}
```

```
void mostrar () {  
    mostrar (0);  
}  
void mostrar (int i) {  
    if (i < 4) {  
        print(i);  
        mostrar (i + 1);  
    }  
}
```

Tela

true

i	0
---	---

Exemplo de Método Iterativo vs Recursivo

- Faça métodos ITERATIVO e RECURSIVO para mostrar os números 0 à 3

```
void mostrar (){
    for (int i = 0; i < 4; i = i + 1) {
        print(i);
    }
}
```

```
void mostrar () {
    mostrar (0);
}
void mostrar (int i){
    if (i < 4) {
        print(i);
        mostrar (i + 1);
    }
}
```

Tela
0

i	0
---	---

Exemplo de Método Iterativo vs Recursivo

- Faça métodos ITERATIVO e RECURSIVO para mostrar os números 0 à 3

```
void mostrar () {  
    for (int i = 0; i < 4; i = i + 1) {  
        print(i);  
    }  
}
```

Tela
0

```
void mostrar () {  
    mostrar (0);  
}  
void mostrar (int i) {  
    if (i < 4) {  
        print(i);  
        mostrar (i + 1);  
    }  
}
```

i	1
---	---

Exemplo de Método Iterativo vs Recursivo

- Faça métodos ITERATIVO e RECURSIVO para mostrar os números 0 à 3

```
void mostrar () {  
    for (int i = 0; i < 4; i = i + 1) {  
        print(i);  
    }  
}
```

Tela
0

```
void mostrar () {  
    mostrar (0);  
}  
void mostrar (int i) {  
    if (i < 4) {  
        print(i);  
        mostrar (i + 1);  
    }  
}
```

i	1
---	---

Exemplo de Método Iterativo vs Recursivo

- Faça métodos ITERATIVO e RECURSIVO para mostrar os números 0 à 3

```
void mostrar () {  
    for (int i = 0; i < 4; i = i + 1) {  
        print(i);  
    }  
}
```

```
void mostrar () {  
    mostrar (0);  
}  
void mostrar (int i) {  
    if (i < 4) {  
        print(i);  
        mostrar (i + 1);  
    }  
}
```

Tela
0

true

i	1
---	---

Exemplo de Método Iterativo vs Recursivo

- Faça métodos ITERATIVO e RECURSIVO para mostrar os números 0 à 3

```
void mostrar (){
    for (int i = 0; i < 4; i = i + 1) {
        print(i);
    }
}
```

```
void mostrar () {
    mostrar (0);
}
void mostrar (int i){
    if (i < 4) {
        print(i);
        mostrar (i + 1);
    }
}
```

Tela
0
1

i	1
---	---

Exemplo de Método Iterativo vs Recursivo

- Faça métodos ITERATIVO e RECURSIVO para mostrar os números 0 à 3

```
void mostrar () {  
    for (int i = 0; i < 4; i = i + 1) {  
        print(i);  
    }  
}
```

Tela
0
1

```
void mostrar () {  
    mostrar (0);  
}  
void mostrar (int i) {  
    if (i < 4) {  
        print(i);  
        mostrar (i + 1);  
    }  
}
```

i	2
---	---

Exemplo de Método Iterativo vs Recursivo

- Faça métodos ITERATIVO e RECURSIVO para mostrar os números 0 à 3

```
void mostrar () {  
    for (int i = 0; i < 4; i = i + 1) {  
        print(i);  
    }  
}
```

Tela
0
1

```
void mostrar () {  
    mostrar (0);  
}  
void mostrar (int i) {  
    if (i < 4) {  
        print(i);  
        mostrar (i + 1);  
    }  
}
```

i	2
---	---

Exemplo de Método Iterativo vs Recursivo

- Faça métodos ITERATIVO e RECURSIVO para mostrar os números 0 à 3

```
void mostrar () {  
    for (int i = 0; i < 4; i = i + 1) {  
        print(i);  
    }  
}
```

```
void mostrar () {  
    mostrar (0);  
}  
void mostrar (int i) {  
    if (i < 4) {  
        print(i);  
        mostrar (i + 1);  
    }  
}
```

Tela
0
1

true

i	2
---	---

Exemplo de Método Iterativo vs Recursivo

- Faça métodos ITERATIVO e RECURSIVO para mostrar os números 0 à 3

```
void mostrar (){
    for (int i = 0; i < 4; i = i + 1) {
        print(i);
    }
}
```

```
void mostrar () {
    mostrar (0);
}
void mostrar (int i){
    if (i < 4) {
        print(i);
        mostrar (i + 1);
    }
}
```

Tela
0
1
2

i	2
---	---

Exemplo de Método Iterativo vs Recursivo

- Faça métodos ITERATIVO e RECURSIVO para mostrar os números 0 à 3

```
void mostrar () {  
    for (int i = 0; i < 4; i = i + 1) {  
        print(i);  
    }  
}
```

Tela
0
1
2

```
void mostrar () {  
    mostrar (0);  
}  
void mostrar (int i) {  
    if (i < 4) {  
        print(i);  
        mostrar (i + 1);  
    }  
}
```

i	3
---	---

Exemplo de Método Iterativo vs Recursivo

- Faça métodos ITERATIVO e RECURSIVO para mostrar os números 0 à 3

```
void mostrar () {  
    for (int i = 0; i < 4; i = i + 1) {  
        print(i);  
    }  
}
```

Tela
0
1
2

```
void mostrar () {  
    mostrar (0);  
}  
void mostrar (int i) {  
    if (i < 4) {  
        print(i);  
        mostrar (i + 1);  
    }  
}
```

i	3
---	---

Exemplo de Método Iterativo vs Recursivo

- Faça métodos ITERATIVO e RECURSIVO para mostrar os números 0 à 3

```
void mostrar () {  
    for (int i = 0; i < 4; i = i + 1) {  
        print(i);  
    }  
}
```

```
void mostrar () {  
    mostrar (0);  
}  
void mostrar (int i) {  
    if (i < 4) {  
        print(i);  
        mostrar (i + 1);  
    }  
}
```

Tela
0
1
2

true

i	3
---	---

Exemplo de Método Iterativo vs Recursivo

- Faça métodos ITERATIVO e RECURSIVO para mostrar os números 0 à 3

```
void mostrar (){
    for (int i = 0; i < 4; i = i + 1) {
        print(i);
    }
}
```

```
void mostrar () {
    mostrar (0);
}
void mostrar (int i){
    if (i < 4) {
        print(i);
        mostrar (i + 1);
    }
}
```

Tela
0
1
2
3

i	3
---	---

Exemplo de Método Iterativo vs Recursivo

- Faça métodos ITERATIVO e RECURSIVO para mostrar os números 0 à 3

```
void mostrar (){
    for (int i = 0; i < 4; i = i + 1) {
        print(i);
    }
}
```

Tela
0
1
2
3

```
void mostrar () {
    mostrar (0);
}
void mostrar (int i){
    if (i < 4) {
        print(i);
        mostrar (i + 1);
    }
}
```

i	4
---	---

Exemplo de Método Iterativo vs Recursivo

- Faça métodos ITERATIVO e RECURSIVO para mostrar os números 0 à 3

```
void mostrar () {  
    for (int i = 0; i < 4; i = i + 1) {  
        print(i);  
    }  
}
```

Tela
0
1
2
3

```
void mostrar () {  
    mostrar (0);  
}  
void mostrar (int i) {  
    if (i < 4) {  
        print(i);  
        mostrar (i + 1);  
    }  
}
```

i	4
---	---

Exemplo de Método Iterativo vs Recursivo

- Faça métodos ITERATIVO e RECURSIVO para mostrar os números 0 à 3

```
void mostrar () {
    for (int i = 0; i < 4; i = i + 1) {
        print(i);
    }
}
```

```
void mostrar () {
    mostrar (0);
}

void mostrar (int i) {
    if (i < 4) {
        print(i);
        mostrar (i + 1);
    }
}
```

false

Tela
0
1
2
3

i	4
---	---

- Por que o código abaixo imprime 2, 1, 0, 0, 1 e 2?

```
void printRecursivo(){  
    printRecursivo(2);  
}  
  
void printRecursivo(int i){  
    print(i);  
    if (i > 0){  
        printRecursivo(i - 1);  
    }  
    print(i);  
}
```

- Por que o código abaixo imprime 2, 1, 0, 0, 1 e 2?

```
void printRecursivo(){  
    printRecursivo(2);  
}  
  
void printRecursivo(int i){  
    print(i);  
    if (i > 0){  
        printRecursivo(i - 1);  
    }  
    print(i);  
}
```

É fácil compreender a impressão do 2, 1, 0 e 0. Contudo, o 1 e 2 no final ... 😞

Exercício

- Por que o código abaixo imprime 2, 1, 0, 0, 1 e 2?

```
void printRecursivo(){  
    printRecursivo(2);  
}  
  
void printRecursivo(int i){  
    print(i);  
    if (i > 0){  
        printRecursivo(i - 1);  
    }  
    print(i);  
}
```

É fácil compreender a impressão do 2, 1, 0 e 0. Contudo, o 1 e 2 no final ... 😞

Tela

i	
---	--

Exercício

- Por que o código abaixo imprime 2, 1, 0, 0, 1 e 2?

```
void printRecursivo(){  
    printRecursivo(2);  
}  
  
void printRecursivo(int i){  
    print(i);  
    if (i > 0){  
        printRecursivo(i - 1);  
    }  
    print(i);  
}
```

É fácil compreender a impressão do 2, 1, 0 e 0. Contudo, o 1 e 2 no final ... 😞

Tela

i	
---	--

Exercício

- Por que o código abaixo imprime 2, 1, 0, 0, 1 e 2?

```
void printRecursivo(){  
    printRecursivo(2);  
}
```

```
void printRecursivo(int i){ // i (2)  
    print(i);  
    if (i > 0){  
        printRecursivo(i - 1);  
    }  
    print(i);  
}
```

É fácil compreender a impressão do 2, 1, 0 e 0. Contudo, o 1 e 2 no final ... ☹

Tela

i	2
---	---

Exercício

- Por que o código abaixo imprime 2, 1, 0, 0, 1 e 2?

```
void printRecursivo(){  
    printRecursivo(2);  
}  
  
void printRecursivo(int i){ // i (2)  
    print(i);  
    if (i > 0){  
        printRecursivo(i - 1);  
    }  
    print(i);  
}
```

É fácil compreender a impressão do 2, 1, 0 e 0. Contudo, o 1 e 2 no final ... 😞

Tela

2

i	2
---	---

Exercício

- Por que o código abaixo imprime 2, 1, 0, 0, 1 e 2?

```
void printRecursivo(){  
    printRecursivo(2);  
}  
  
void printRecursivo(int i){ // i (2)  
    print(i);  
    if (i > 0){  
        printRecursivo(i - 1);  
    }  
    print(i);  
}
```

É fácil compreender a impressão do 2, 1, 0 e 0. Contudo, o 1 e 2 no final ... ☹

Tela

2

i	2
---	---

Exercício

- Por que o código abaixo imprime 2, 1, 0, 0, 1 e 2?

```
void printRecursivo(){  
    printRecursivo(2);  
}  
  
void printRecursivo(int i){ // i (2)  
    print(i);  
    if (i > 0){  
        printRecursivo(i - 1);  
    }  
    print(i);  
}
```

É fácil compreender a impressão do 2, 1, 0 e 0. Contudo, o 1 e 2 no final ... 😞

Tela

2

i	2
---	---

Exercício

- Por que o código abaixo imprime 2, 1, 0, 0, 1 e 2?

```
void printRecursivo(){  
    printRecursivo(2);  
}
```

```
void printRecursivo(int i){ // i (1)  
    print(i);  
    if (i > 0){  
        printRecursivo(i - 1);  
    }  
    print(i);  
}
```

É fácil compreender a impressão do 2, 1, 0 e 0. Contudo, o 1 e 2 no final ... ☹

Tela

2

i	1
---	---

Exercício

- Por que o código abaixo imprime 2, 1, 0, 0, 1 e 2?

```
void printRecursivo(){  
    printRecursivo(2);  
}  
  
void printRecursivo(int i){ // i (1)  
    print(i);  
    if (i > 0){  
        printRecursivo(i - 1);  
    }  
    print(i);  
}
```

É fácil compreender a impressão do 2, 1, 0 e 0. Contudo, o 1 e 2 no final ... ☹

Tela

2
1

i	1
---	---

Exercício

- Por que o código abaixo imprime 2, 1, 0, 0, 1 e 2?

```
void printRecursivo(){  
    printRecursivo(2);  
}  
  
void printRecursivo(int i){ // i (1)  
    print(i);  
    if (i > 0){  
        printRecursivo(i - 1);  
    }  
    print(i);  
}
```

É fácil compreender a impressão do 2, 1, 0 e 0. Contudo, o 1 e 2 no final ... ☹

Tela

2
1

i	1
---	---

Exercício

- Por que o código abaixo imprime 2, 1, 0, 0, 1 e 2?

```
void printRecursivo(){  
    printRecursivo(2);  
}  
  
void printRecursivo(int i){ // i (1)  
    print(i);  
    if (i > 0){  
        printRecursivo(i - 1);  
    }  
    print(i);  
}
```

É fácil compreender a impressão do 2, 1, 0 e 0. Contudo, o 1 e 2 no final ... ☹

Tela
2
1

i	1
---	---

Exercício

- Por que o código abaixo imprime 2, 1, 0, 0, 1 e 2?

```
void printRecursivo(){  
    printRecursivo(2);  
}
```

```
void printRecursivo(int i){ // i (0)  
    print(i);  
    if (i > 0){  
        printRecursivo(i - 1);  
    }  
    print(i);  
}
```

É fácil compreender a impressão do 2, 1, 0 e 0. Contudo, o 1 e 2 no final ... ☹

Tela

2
1

i	0
---	---

Exercício

- Por que o código abaixo imprime 2, 1, 0, 0, 1 e 2?

```
void printRecursivo(){  
    printRecursivo(2);  
}  
  
void printRecursivo(int i){ // i (0)  
    print(i);  
    if (i > 0){  
        printRecursivo(i - 1);  
    }  
    print(i);  
}
```

É fácil compreender a impressão do 2, 1, 0 e 0. Contudo, o 1 e 2 no final ... ☹

Tela
2
1
0

i	0
---	---

Exercício

- Por que o código abaixo imprime 2, 1, 0, 0, 1 e 2?

```
void printRecursivo(){  
    printRecursivo(2);  
}  
  
void printRecursivo(int i){ // i (0)  
    print(i);  
    if (i > 0){  
        printRecursivo(i - 1);  
    }  
    print(i);  
}
```

false

É fácil compreender a impressão do 2, 1, 0 e 0. Contudo, o 1 e 2 no final ... ☹

Tela

2
1
0

i	0
---	---

Exercício

- Por que o código abaixo imprime 2, 1, 0, 0, 1 e 2?

```
void printRecursivo(){  
    printRecursivo(2);  
}  
  
void printRecursivo(int i){ // i (0)  
    print(i);  
    if (i > 0){  
        printRecursivo(i - 1);  
    }  
    print(i);  
}
```

E agora José?

Tela

2
1
0
0

i	0
---	---

Exercício

- Por que o código abaixo imprime 2, 1, 0, 0, 1 e 2?

```
void printRecursivo(){  
    printRecursivo(2);  
}  
  
void printRecursivo(int i){ // i (0)  
    print(i);  
    if (i > 0){  
        printRecursivo(i - 1);  
    }  
    print(i);  
}
```

Agora, estamos no print do zero e retornaremos para o do um onde i vale 1 e chamou o print do zero

Tela
2
1
0
0

i	0
---	---

Exercício

- Por que o código abaixo imprime 2, 1, 0, 0, 1 e 2?

```
void printRecursivo(){  
    printRecursivo(2);  
}  
  
void printRecursivo(int i){ // i (1)  
    print(i);  
    if (i > 0){  
        printRecursivo(i - 1);  
    }  
    print(i);  
}
```

Agora, estamos no print do zero e retornaremos para o do um onde i vale 1 e chamou o print do zero

Tela

2
1
0
0

i	1
---	---

Exercício

- Por que o código abaixo imprime 2, 1, 0, 0, 1 e 2?

```
void printRecursivo(){  
    printRecursivo(2);  
}  
  
void printRecursivo(int i){ // i (1)  
    print(i);  
    if (i > 0){  
        printRecursivo(i - 1);  
    }  
    print(i);  
}
```

Agora, retornaremos para o print do dois

Tela

2
1
0
0
1

i	1
---	---

- Por que o código abaixo imprime 2, 1, 0, 0, 1 e 2?

```
void printRecursivo(){  
    printRecursivo(2);  
}  
  
void printRecursivo(int i){ // i (2)  
    print(i);  
    if (i > 0){  
        printRecursivo(i - 1);  
    }  
    print(i);  
}
```

Agora, retornaremos para o print do dois

Tela

2
1
0
0
1

i	2
---	---

- Por que o código abaixo imprime 2, 1, 0, 0, 1 e 2?

```
void printRecursivo(){  
    printRecursivo(2);  
}  
  
void printRecursivo(int i){ // i (2)  
    print(i);  
    if (i > 0){  
        printRecursivo(i - 1);  
    }  
    print(i);  
}
```

Agora, retornaremos para o primeiro print

Tela

2
1
0
0
1
2

i	2
---	---

Exercício

- Por que o código abaixo imprime 2, 1, 0, 0, 1 e 2?

```
void printRecursivo(){  
    printRecursivo(2);  
}  
  
void printRecursivo(int i){ // i (2)  
    print(i);  
    if (i > 0){  
        printRecursivo(i - 1);  
    }  
    print(i);  
}
```

Agora, retornaremos para o primeiro print

Tela

2
1
0
0
1
2

i	
---	--

Exemplo: Fatorial Recursivo

- Definição do fatorial é recursiva:

$$\begin{cases} \text{Fat}(1) = 1 \\ \text{Fat}(n) = n * \text{Fat}(n - 1) \end{cases}$$

Exemplo: Fatorial Recursivo

- Definição do fatorial é recursiva:

$$\begin{cases} \text{Fat}(1) = 1 \\ \text{Fat}(n) = n * \text{Fat}(n - 1) \end{cases}$$

- Qual é o valor do fatorial de 5?

Exemplo: Fatorial Recursivo

- Definição do fatorial é recursiva:

$$\begin{cases} \text{Fat}(1) = 1 \\ \text{Fat}(n) = n * \text{Fat}(n - 1) \end{cases}$$

- Qual é o valor do fatorial de 5?

$$\text{Fat}(5) = 5 * \text{Fat}(4)$$


Exemplo: Fatorial Recursivo

- Definição do fatorial é recursiva:

$$\begin{cases} \text{Fat}(1) = 1 \\ \text{Fat}(n) = n * \text{Fat}(n - 1) \end{cases}$$

$\text{Fat}(4) = 4 * \text{Fat}(3)$

- Qual é o valor do fatorial de 5?

$$\text{Fat}(5) = 5 * \text{Fat}(4)$$


Exemplo: Fatorial Recursivo

- Definição do fatorial é recursiva:

$$\begin{cases} \text{Fat}(1) = 1 \\ \text{Fat}(n) = n * \text{Fat}(n - 1) \end{cases}$$

$\text{Fat}(4) = 4 * \text{Fat}(3)$

$\text{Fat}(3) = 3 * \text{Fat}(2)$

- Qual é o valor do fatorial de 5?

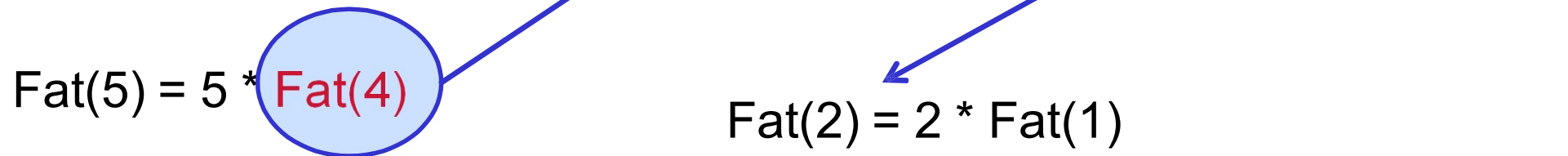
$\text{Fat}(5) = 5 * \text{Fat}(4)$

Exemplo: Fatorial Recursivo

- Definição do fatorial é recursiva:

$$\begin{cases} \text{Fat}(1) = 1 \\ \text{Fat}(n) = n * \text{Fat}(n - 1) \end{cases}$$

- Qual é o valor do fatorial de 5?

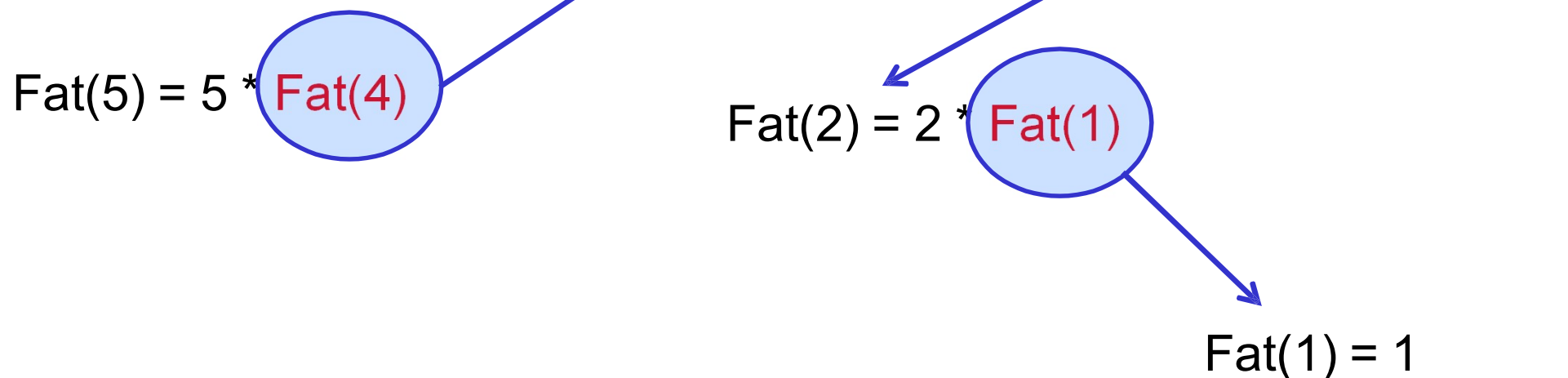


Exemplo: Fatorial Recursivo

- Definição do fatorial é recursiva:

$$\begin{cases} \text{Fat}(1) = 1 \\ \text{Fat}(n) = n * \text{Fat}(n - 1) \end{cases}$$

- Qual é o valor do fatorial de 5?

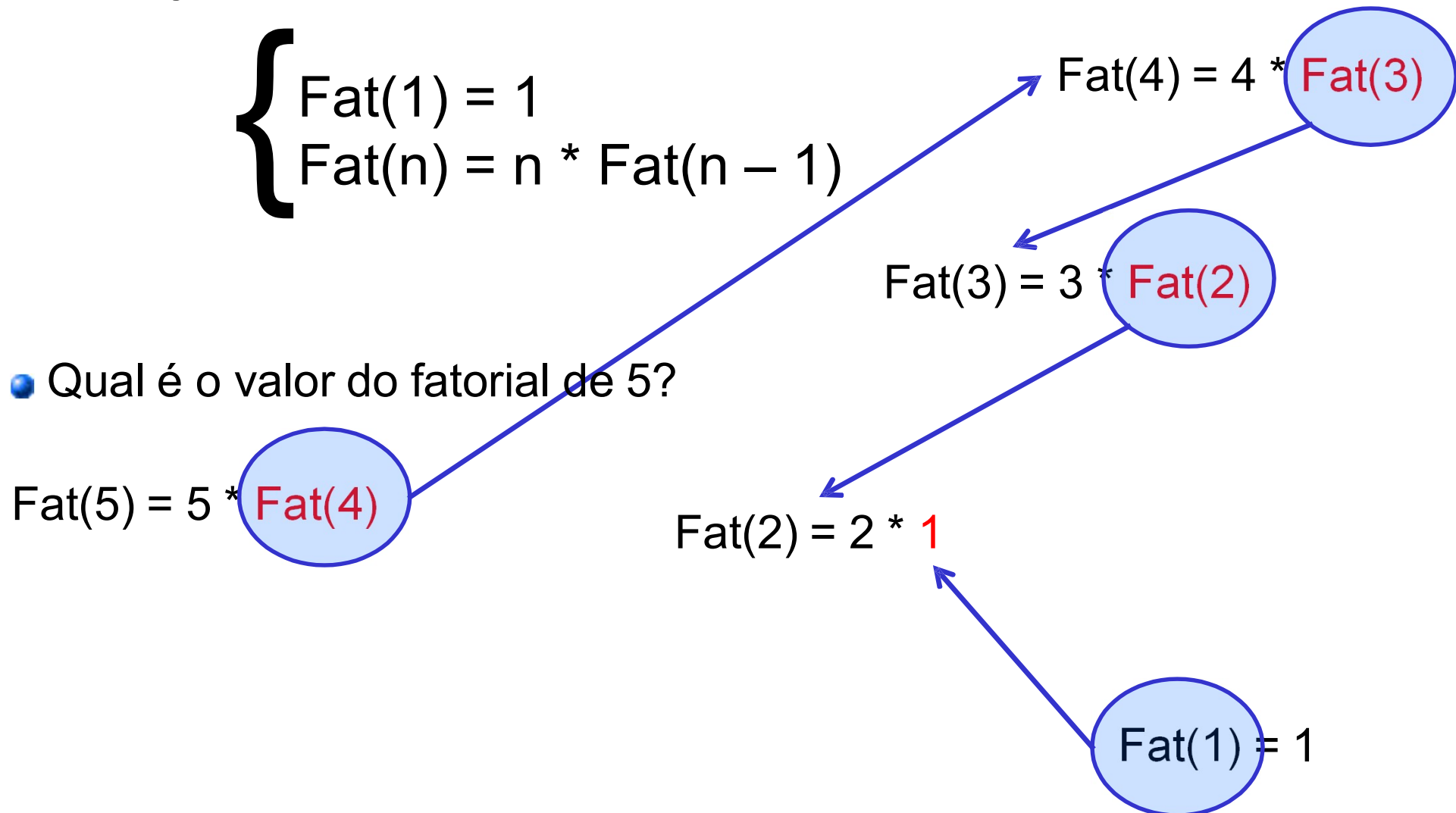


Exemplo: Fatorial Recursivo

- Definição do fatorial é recursiva:

$$\begin{cases} \text{Fat}(1) = 1 \\ \text{Fat}(n) = n * \text{Fat}(n - 1) \end{cases}$$

- Qual é o valor do fatorial de 5?

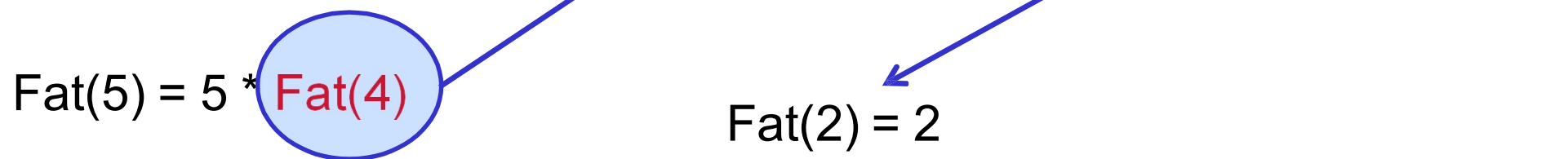


Exemplo: Fatorial Recursivo

- Definição do fatorial é recursiva:

$$\begin{cases} \text{Fat}(1) = 1 \\ \text{Fat}(n) = n * \text{Fat}(n - 1) \end{cases}$$

- Qual é o valor do fatorial de 5?

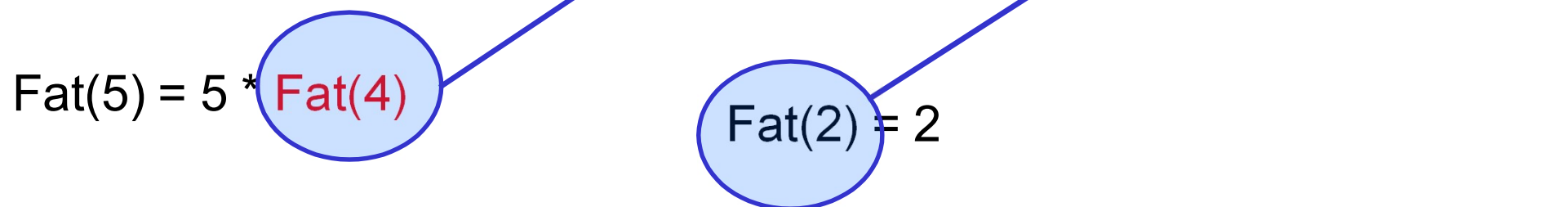


Exemplo: Fatorial Recursivo

- Definição do fatorial é recursiva:

$$\begin{cases} \text{Fat}(1) = 1 \\ \text{Fat}(n) = n * \text{Fat}(n - 1) \end{cases}$$

- Qual é o valor do fatorial de 5?



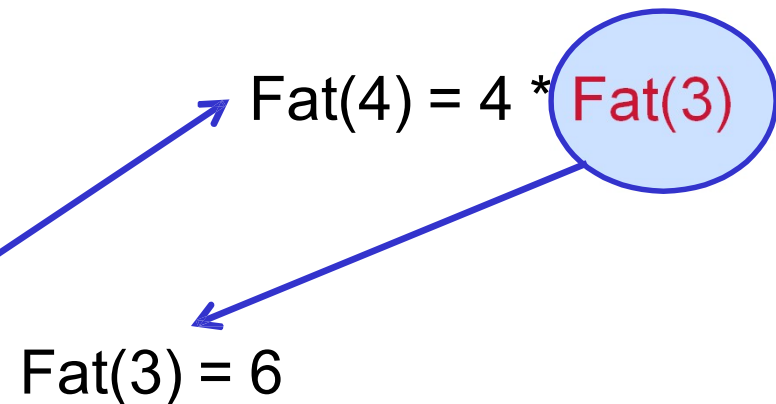
Exemplo: Fatorial Recursivo

- Definição do fatorial é recursiva:

$$\begin{cases} \text{Fat}(1) = 1 \\ \text{Fat}(n) = n * \text{Fat}(n - 1) \end{cases}$$

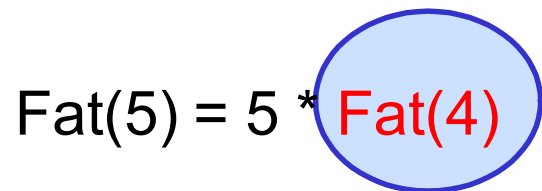
$\text{Fat}(4) = 4 * \text{Fat}(3)$

$\text{Fat}(3) = 6$



- Qual é o valor do fatorial de 5?

$\text{Fat}(5) = 5 * \text{Fat}(4)$



Exemplo: Fatorial Recursivo

- Definição do fatorial é recursiva:

$$\begin{cases} \text{Fat}(1) = 1 \\ \text{Fat}(n) = n * \text{Fat}(n - 1) \end{cases}$$

- Qual é o valor do fatorial de 5?

$$\text{Fat}(5) = 5 * \text{Fat}(4)$$

$$\text{Fat}(3) = 6$$

$$\text{Fat}(4) = 4 * 6$$

Exemplo: Fatorial Recursivo

- Definição do fatorial é recursiva:

$$\begin{cases} \text{Fat}(1) = 1 \\ \text{Fat}(n) = n * \text{Fat}(n - 1) \end{cases}$$

$$\text{Fat}(4) = 24$$

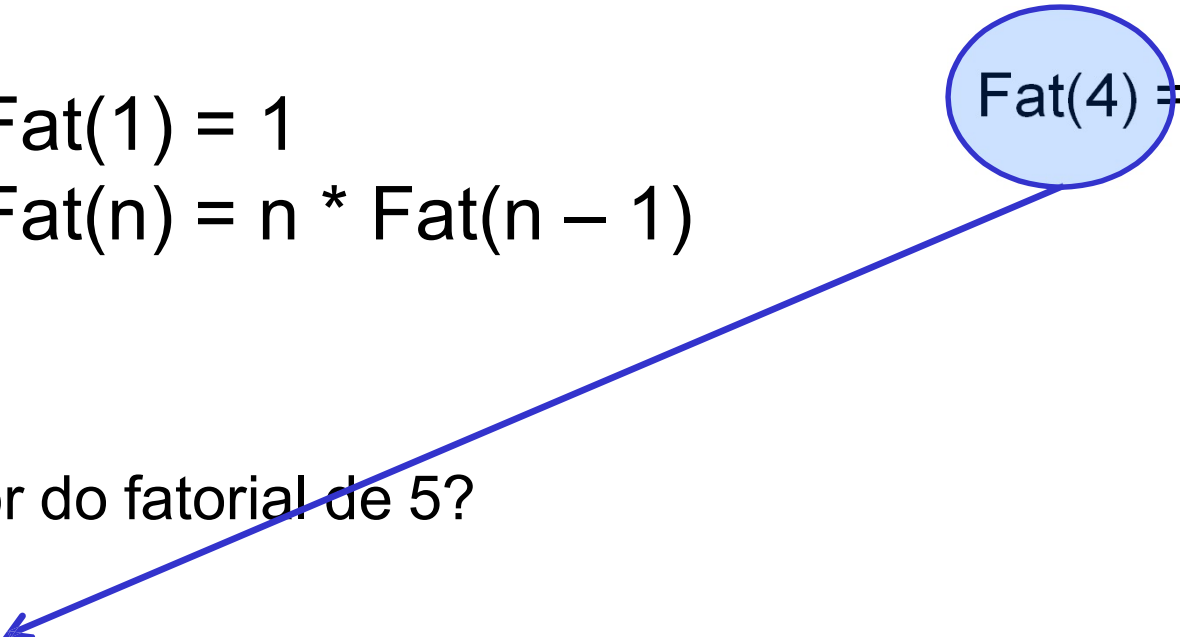
- Qual é o valor do fatorial de 5?

$$\text{Fat}(5) = 5 * \text{Fat}(4)$$

Exemplo: Fatorial Recursivo

- Definição do fatorial é recursiva:

$$\begin{cases} \text{Fat}(1) = 1 \\ \text{Fat}(n) = n * \text{Fat}(n - 1) \end{cases}$$


$$\text{Fat}(4) = 24$$

- Qual é o valor do fatorial de 5?

$$\text{Fat}(5) = 5 * 24$$

Exemplo: Fatorial Recursivo

- Definição do fatorial é recursiva:

$$\begin{cases} \text{Fat}(1) = 1 \\ \text{Fat}(n) = n * \text{Fat}(n - 1) \end{cases}$$

- Qual é o valor do fatorial de 5?

$\text{Fat}(5) = 120$

Exemplo: Fatorial Recursivo

```
int fatorial (int n){  
    int resp;  
    if (n == 1){  
        resp = 1;  
    } else {  
        resp = n * fatorial(n - 1);  
    }  
    return resp;  
}  
void main(){  
    int valor = fatorial(5);  
    print(valor);  
}
```

Exemplo: Fatorial Recursivo

```
int fatorial (int n){  
    int resp;  
    if (n == 1){  
        resp = 1;  
    } else {  
        resp = n * fatorial(n - 1);  
    }  
    return resp;  
}  
  
void main(){  
    int valor = fatorial(5);  
    print(valor);  
}
```

Exemplo: Fatorial Recursivo

```
int fatorial (int n){  
    int resp;  
    if (n == 1){  
        resp = 1;  
    } else {  
        resp = n * fatorial(n - 1);  
    }  
    return resp;  
}  
void main(){  
    int valor = fatorial(5);  
    print(valor);  
}
```

Exemplo: Fatorial Recursivo

```
int fatorial (int n){ // n (5)
    int resp;
    if (n == 1){
        resp = 1;
    } else {
        resp = n * fatorial(n - 1);
    }
    return resp;
}
void main(){
    int valor = fatorial(5);
    print(valor);
}
```

Exemplo: Fatorial Recursivo

```
int fatorial (int n){ // n (5)
    int resp;
    if (n == 1){
        resp = 1;
    } else {
        resp = n * fatorial(n - 1);
    }
    return resp;
}

void main(){
    int valor = fatorial(5);
    print(valor);
}
```

Exemplo: Fatorial Recursivo

```
int fatorial (int n){ // n (5)
    int resp;
    if (n == 1){
        resp = 1;
    } else {
        resp = n * fatorial(n - 1);
    }
    return resp;
}

void main(){
    int valor = fatorial(5);
    print(valor);
}
```

Exemplo: Fatorial Recursivo

```
int fatorial (int n){ // n (5)
    int resp;
    if (n == 1){
        resp = 1;
    } else {
        resp = n * fatorial(n - 1);
    }
    return resp;
}

void main(){
    int valor = fatorial(5);
    print(valor);
}
```

Exemplo: Fatorial Recursivo

```
int fatorial (int n){ // n (5)
    int resp;
    if (n == 1){
        resp = 1;
    } else {
        resp = n * fatorial(n - 1);
    }
    return resp;
}

void main(){
    int valor = fatorial(5);
    print(valor);
}
```

$\text{fatorial}(5) = 5 * \text{fatorial}(4)$

Exemplo: Fatorial Recursivo

```
int fatorial (int n){ // n (4)
    int resp;
    if (n == 1){
        resp = 1;
    } else {
        resp = n * fatorial(n - 1);
    }
    return resp;
}

void main(){
    int valor = fatorial(5);
    print(valor);
}
```

fatorial (5) = 5 * fatorial (4)

fatorial (4) =



Exemplo: Fatorial Recursivo

```
int fatorial (int n){ // n (4)
    int resp;
    if (n == 1){
        resp = 1;
    } else {
        resp = n * fatorial(n - 1);
    }
    return resp;
}

void main(){
    int valor = fatorial(5);
    print(valor);
}
```

fatorial (5) = 5 * fatorial (4)

fatorial (4) =



Exemplo: Fatorial Recursivo

```
int fatorial (int n){ // n (4)
    int resp;
    if (n == 1){
        resp = 1;
    } else {
        resp = n * fatorial(n - 1);
    }
    return resp;
}

void main(){
    int valor = fatorial(5);
    print(valor);
}
```

fatorial (5) = 5 * fatorial (4)

fatorial (4) =



Exemplo: Fatorial Recursivo

```
int fatorial (int n){ // n (4)
    int resp;
    if (n == 1){
        resp = 1;
    } else {
        resp = n * fatorial(n - 1);
    }
    return resp;
}

void main(){
    int valor = fatorial(5);
    print(valor);
}
```

fatorial (5) = 5 * fatorial (4)

fatorial (4) =



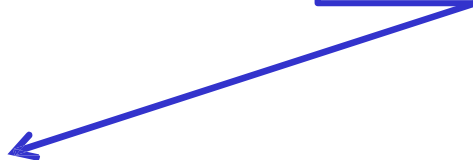
Exemplo: Fatorial Recursivo

```
int fatorial (int n){ // n (4)
    int resp;
    if (n == 1){
        resp = 1;
    } else {
        resp = n * fatorial(n - 1);
    }
    return resp;
}

void main(){
    int valor = fatorial(5);
    print(valor);
}
```

fatorial (5) = 5 * fatorial (4)

fatorial (4) = 4 * fatorial (3)



Exemplo: Fatorial Recursivo

```
int fatorial (int n){ // n (3)
    int resp;
    if (n == 1){
        resp = 1;
    } else {
        resp = n * fatorial(n - 1);
    }
    return resp;
}

void main(){
    int valor = fatorial(5);
    print(valor);
}
```

fatorial (5) = 5 * fatorial (4)

fatorial (4) = 4 * fatorial (3)

fatorial (3) =

Exemplo: Fatorial Recursivo

```
int fatorial (int n){ // n (3)
    int resp;
    if (n == 1){
        resp = 1;
    } else {
        resp = n * fatorial(n - 1);
    }
    return resp;
}

void main(){
    int valor = fatorial(5);
    print(valor);
}
```

fatorial (5) = 5 * fatorial (4)

fatorial (4) = 4 * fatorial (3)

fatorial (3) =

Exemplo: Fatorial Recursivo

```
int fatorial (int n){ // n (3)
    int resp;
    if (n == 1){
        resp = 1;
    } else {
        resp = n * fatorial(n - 1);
    }
    return resp;
}

void main(){
    int valor = fatorial(5);
    print(valor);
}
```

fatorial (5) = 5 * fatorial (4)

fatorial (4) = 4 * fatorial (3)

fatorial (3) =

Exemplo: Fatorial Recursivo

```
int fatorial (int n){ // n (3)
    int resp;
    if (n == 1){
        resp = 1;
    } else {
        resp = n * fatorial(n - 1);
    }
    return resp;
}

void main(){
    int valor = fatorial(5);
    print(valor);
}
```

fatorial (5) = 5 * fatorial (4)

fatorial (4) = 4 * fatorial (3)

fatorial (3) =

Exemplo: Fatorial Recursivo

```
int fatorial (int n){ // n (3)
    int resp;
    if (n == 1){
        resp = 1;
    } else {
        resp = n * fatorial(n - 1);
    }
    return resp;
}

void main(){
    int valor = fatorial(5);
    print(valor);
}
```

fatorial (5) = 5 * fatorial (4)

fatorial (4) = 4 * fatorial (3)

fatorial (3) = 3 * fatorial (2)

Exemplo: Fatorial Recursivo

```
int fatorial (int n){ // n (2)
    int resp;
    if (n == 1){
        resp = 1;
    } else {
        resp = n * fatorial(n - 1);
    }
    return resp;
}

void main(){
    int valor = fatorial(5);
    print(valor);
}
```

fatorial (5) = 5 * fatorial (4)

fatorial (4) = 4 * fatorial (3)

fatorial (3) = 3 * fatorial (2)

fatorial (2) =

Exemplo: Fatorial Recursivo

```
int fatorial (int n){ // n (2)
    int resp;
    if (n == 1){
        resp = 1;
    } else {
        resp = n * fatorial(n - 1);
    }
    return resp;
}

void main(){
    int valor = fatorial(5);
    print(valor);
}
```

fatorial (5) = 5 * fatorial (4)

fatorial (4) = 4 * fatorial (3)

fatorial (3) = 3 * fatorial (2)

fatorial (2) =

Exemplo: Fatorial Recursivo

```
int fatorial (int n){ // n (2)
    int resp;
    if (n == 1){
        resp = 1;
    } else {
        resp = n * fatorial(n - 1);
    }
    return resp;
}

void main(){
    int valor = fatorial(5);
    print(valor);
}
```

fatorial (5) = 5 * fatorial (4)

fatorial (4) = 4 * fatorial (3)

fatorial (3) = 3 * fatorial (2)

fatorial (2) =

Exemplo: Fatorial Recursivo

```
int fatorial (int n){ // n (2)
    int resp;
    if (n == 1){
        resp = 1;
    } else {
        resp = n * fatorial(n - 1);
    }
    return resp;
}

void main(){
    int valor = fatorial(5);
    print(valor);
}
```

fatorial (5) = 5 * fatorial (4)

fatorial (4) = 4 * fatorial (3)

fatorial (3) = 3 * fatorial (2)

fatorial (2) =

Exemplo: Fatorial Recursivo

```
int fatorial (int n){ // n (2)
    int resp;
    if (n == 1){
        resp = 1;
    } else {
        resp = n * fatorial(n - 1);
    }
    return resp;
}

void main(){
    int valor = fatorial(5);
    print(valor);
}
```

fatorial (5) = 5 * fatorial (4)

fatorial (4) = 4 * fatorial (3)

fatorial (3) = 3 * fatorial (2)

fatorial (2) = 2 * fatorial (1)

Exemplo: Fatorial Recursivo

```
int fatorial (int n){ // n (1)
    int resp;
    if (n == 1){
        resp = 1;
    } else {
        resp = n * fatorial(n - 1);
    }
    return resp;
}

void main(){
    int valor = fatorial(5);
    print(valor);
}
```

fatorial (5) = 5 * fatorial (4)

fatorial (4) = 4 * fatorial (3)

fatorial (3) = 3 * fatorial (2)

fatorial (2) = 2 * fatorial (1)

fatorial (1) =

Exemplo: Fatorial Recursivo

```
int fatorial (int n){ // n (1)
    int resp;
    if (n == 1){
        resp = 1;
    } else {
        resp = n * fatorial(n - 1);
    }
    return resp;
}

void main(){
    int valor = fatorial(5);
    print(valor);
}
```

fatorial (5) = 5 * fatorial (4)

fatorial (4) = 4 * fatorial (3)

fatorial (3) = 3 * fatorial (2)

fatorial (2) = 2 * fatorial (1)

fatorial (1) =

Exemplo: Fatorial Recursivo

```
int fatorial (int n){ // n (1)
    int resp;
    if (n == 1){
        resp = 1;           true
    } else {
        resp = n * fatorial(n - 1);
    }
    return resp;
}

void main(){
    int valor = fatorial(5);
    print(valor);
}
```

fatorial (5) = 5 * fatorial (4)

fatorial (4) = 4 * fatorial (3)

fatorial (3) = 3 * fatorial (2)

fatorial (2) = 2 * fatorial (1)

fatorial (1) =

Exemplo: Fatorial Recursivo

```
int fatorial (int n){ // n (1)
    int resp;
    if (n == 1){
        resp = 1;
    } else {
        resp = n * fatorial(n - 1);
    }
    return resp;
}

void main(){
    int valor = fatorial(5);
    print(valor);
}
```

fatorial (5) = 5 * fatorial (4)

fatorial (4) = 4 * fatorial (3)

fatorial (3) = 3 * fatorial (2)

fatorial (2) = 2 * fatorial (1)

fatorial (1) = 1

Exemplo: Fatorial Recursivo

```
int fatorial (int n){ // n (1)
    int resp;
    if (n == 1){
        resp = 1;
    } else {
        resp = n * fatorial(n - 1);
    }
    return resp;
}

void main(){
    int valor = fatorial(5);
    print(valor);
}
```

fatorial (5) = 5 * fatorial (4)

fatorial (4) = 4 * fatorial (3)

fatorial (3) = 3 * fatorial (2)

fatorial (2) = 2 * fatorial (1)

fatorial (1) = 1

Exemplo: Fatorial Recursivo

```
int fatorial (int n){ // n (2)
    int resp;
    if (n == 1){
        resp = 1;
    } else {
        resp = n * fatorial(n - 1);
    }
    return resp;
}

void main(){
    int valor = fatorial(5);
    print(valor);
}
```

fatorial (5) = 5 * fatorial (4)

fatorial (4) = 4 * fatorial (3)

fatorial (3) = 3 * fatorial (2)

fatorial (2) = 2 * 1

Exemplo: Fatorial Recursivo

```
int fatorial (int n){ // n (2)
    int resp;
    if (n == 1){
        resp = 1;
    } else {
        resp = n * fatorial(n - 1);
    }
    return resp;
}

void main(){
    int valor = fatorial(5);
    print(valor);
}
```

fatorial (5) = 5 * fatorial (4)

fatorial (4) = 4 * fatorial (3)

fatorial (3) = 3 * fatorial (2)

fatorial (2) = 2

Exemplo: Fatorial Recursivo

```
int fatorial (int n){ // n (2)
    int resp;
    if (n == 1){
        resp = 1;
    } else {
        resp = n * fatorial(n - 1);
    }
    return resp;
}

void main(){
    int valor = fatorial(5);
    print(valor);
}
```

fatorial (5) = 5 * fatorial (4)

fatorial (4) = 4 * fatorial (3)

fatorial (3) = 3 * fatorial (2)

fatorial (2) = 2

Exemplo: Fatorial Recursivo

```
int fatorial (int n){ // n (3)
    int resp;
    if (n == 1){
        resp = 1;
    } else {
        resp = n * fatorial(n - 1);
    }
    return resp;
}

void main(){
    int valor = fatorial(5);
    print(valor);
}
```

fatorial (5) = 5 * fatorial (4)

fatorial (4) = 4 * fatorial (3)

fatorial (3) = 3 * 2

Exemplo: Fatorial Recursivo

```
int fatorial (int n){ // n (3)
    int resp;
    if (n == 1){
        resp = 1;
    } else {
        resp = n * fatorial(n - 1);
    }
    return resp;
}

void main(){
    int valor = fatorial(5);
    print(valor);
}
```

fatorial (5) = 5 * fatorial (4)

fatorial (4) = 4 * fatorial (3)

fatorial (3) = 6

Exemplo: Fatorial Recursivo

```
int fatorial (int n){ // n (3)
    int resp;
    if (n == 1){
        resp = 1;
    } else {
        resp = n * fatorial(n - 1);
    }
    return resp;
}

void main(){
    int valor = fatorial(5);
    print(valor);
}
```

fatorial (5) = 5 * fatorial (4)

fatorial (4) = 4 * fatorial (3)

fatorial (3) = 6

Exemplo: Fatorial Recursivo

```
int fatorial (int n){ // n (4)
    int resp;
    if (n == 1){
        resp = 1;
    } else {
        resp = n * fatorial(n - 1);
    }
    return resp;
}

void main(){
    int valor = fatorial(5);
    print(valor);
}
```

fatorial (5) = 5 * fatorial (4)

fatorial (4) = 4 * 6



Exemplo: Fatorial Recursivo

```
int fatorial (int n){ // n (4)
    int resp;
    if (n == 1){
        resp = 1;
    } else {
        resp = n * fatorial(n - 1);
    }
    return resp;
}

void main(){
    int valor = fatorial(5);
    print(valor);
}
```

fatorial (5) = 5 * fatorial (4)

fatorial (4) = 24



Exemplo: Fatorial Recursivo

```
int fatorial (int n){ // n (4)
    int resp;
    if (n == 1){
        resp = 1;
    } else {
        resp = n * fatorial(n - 1);
    }
    return resp;
}

void main(){
    int valor = fatorial(5);
    print(valor);
}
```

fatorial (5) = 5 * fatorial (4)

fatorial (4) = 24



Exemplo: Fatorial Recursivo

```
int fatorial (int n){ // n (5)
    int resp;
    if (n == 1){
        resp = 1;
    } else {
        resp = n * fatorial(n - 1);
    }
    return resp;
}

void main(){
    int valor = fatorial(5);
    print(valor);
}
```

$$\text{fatorial}(5) = 5 * \boxed{24}$$

Exemplo: Fatorial Recursivo

```
int fatorial (int n){ // n (5)
    int resp;
    if (n == 1){
        resp = 1;
    } else {
        resp = n * fatorial(n - 1);
    }
    return resp;
}

void main(){
    int valor = fatorial(5);
    print(valor);
}
```

fatorial (5) = 120

Exemplo: Fatorial Recursivo

```
int fatorial (int n){ // n (5)
    int resp;
    if (n == 1){
        resp = 1;
    } else {
        resp = n * fatorial(n - 1);
    }
    return resp;
}

void main(){
    int valor = fatorial(5);
    print(valor);
}
```

fatorial (5) = 120



Exemplo: Fatorial Recursivo

```
int fatorial (int n){ // n (5)
    int resp;
    if (n == 1){
        resp = 1;
    } else {
        resp = n * fatorial(n - 1);
    }
    return resp;
}

void main(){
    int valor = fatorial(5);
    print(valor);
}
```

fatorial (5) = 120



Exemplo: Fatorial Recursivo

```
int fatorial (int n){ // n (5)
    int resp;
    if (n == 1){
        resp = 1;
    } else {
        resp = n * fatorial(n - 1);
    }
    return resp;
}

void main(){
    int valor = fatorial(5);
    print(valor);
}
```

Exemplo: Fibonacci Recursivo

- Definição do Fibonacci é recursiva:

$$\left\{ \begin{array}{l} \text{Fib}(0) = 1 \\ \text{Fib}(1) = 1 \\ \text{Fib}(n) = \text{Fib}(n-1) + \text{Fib}(n-2) \end{array} \right.$$

- Qual é o Fibonacci de 4?

Exemplo: Fibonacci Recursivo

- Definição do Fibonacci é recursiva:

$$\left\{ \begin{array}{l} \text{Fib}(0) = 1 \\ \text{Fib}(1) = 1 \\ \text{Fib}(n) = \text{Fib}(n-1) + \text{Fib}(n-2) \end{array} \right.$$

- Qual é o Fibonacci de 4? Fibonacci(4)

Exemplo: Fibonacci Recursivo

- Definição do Fibonacci é recursiva:

$$\left\{ \begin{array}{l} \text{Fib}(0) = 1 \\ \text{Fib}(1) = 1 \\ \text{Fib}(n) = \text{Fib}(n-1) + \text{Fib}(n-2) \end{array} \right.$$

- Qual é o Fibonacci de 4?

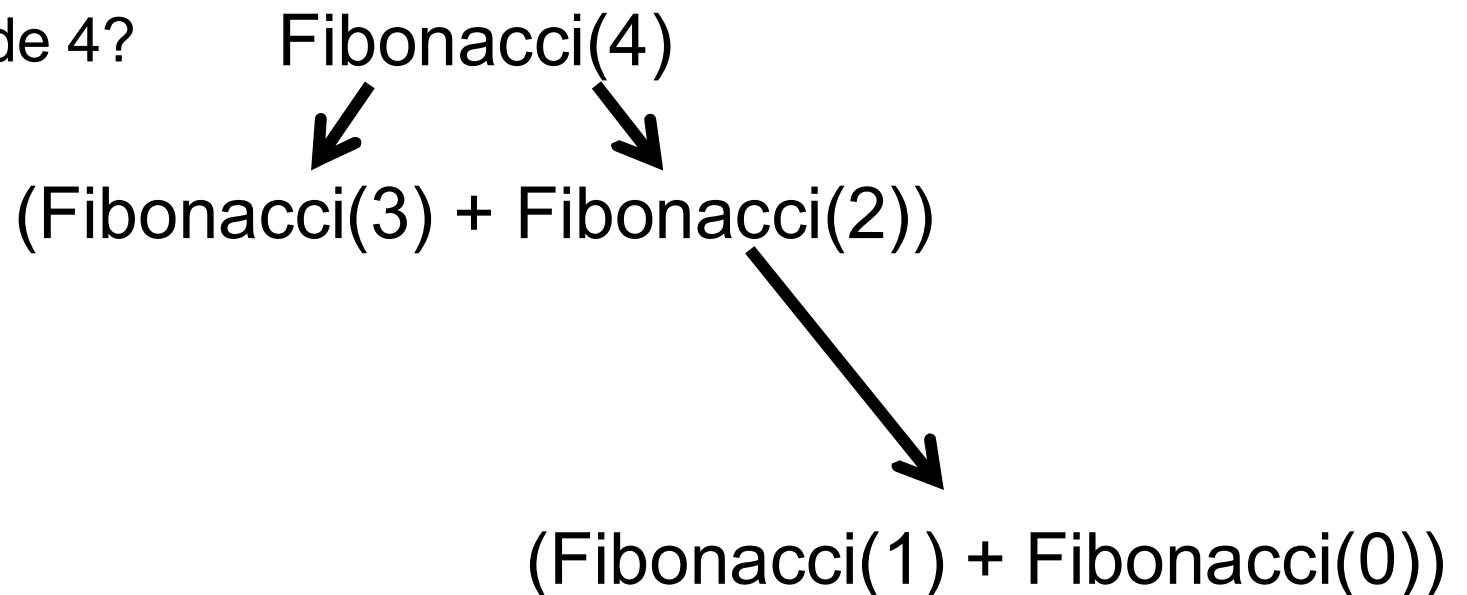
$$\begin{array}{c} \text{Fibonacci}(4) \\ \swarrow \quad \searrow \\ (\text{Fibonacci}(3) + \text{Fibonacci}(2)) \end{array}$$

Exemplo: Fibonacci Recursivo

- Definição do Fibonacci é recursiva:

$$\left\{ \begin{array}{l} \text{Fib}(0) = 1 \\ \text{Fib}(1) = 1 \\ \text{Fib}(n) = \text{Fib}(n-1) + \text{Fib}(n-2) \end{array} \right.$$

- Qual é o Fibonacci de 4?

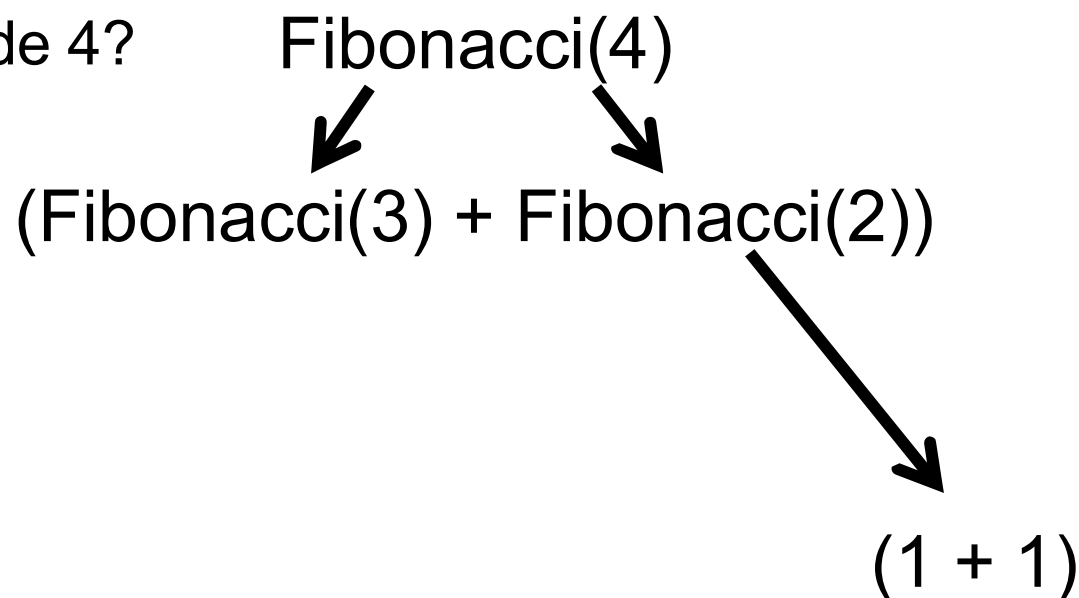


Exemplo: Fibonacci Recursivo

- Definição do Fibonacci é recursiva:

$$\left\{ \begin{array}{l} \text{Fib}(0) = 1 \\ \text{Fib}(1) = 1 \\ \text{Fib}(n) = \text{Fib}(n-1) + \text{Fib}(n-2) \end{array} \right.$$

- Qual é o Fibonacci de 4?



Exemplo: Fibonacci Recursivo

- Definição do Fibonacci é recursiva:

$$\left\{ \begin{array}{l} \text{Fib}(0) = 1 \\ \text{Fib}(1) = 1 \\ \text{Fib}(n) = \text{Fib}(n-1) + \text{Fib}(n-2) \end{array} \right.$$

- Qual é o Fibonacci de 4?

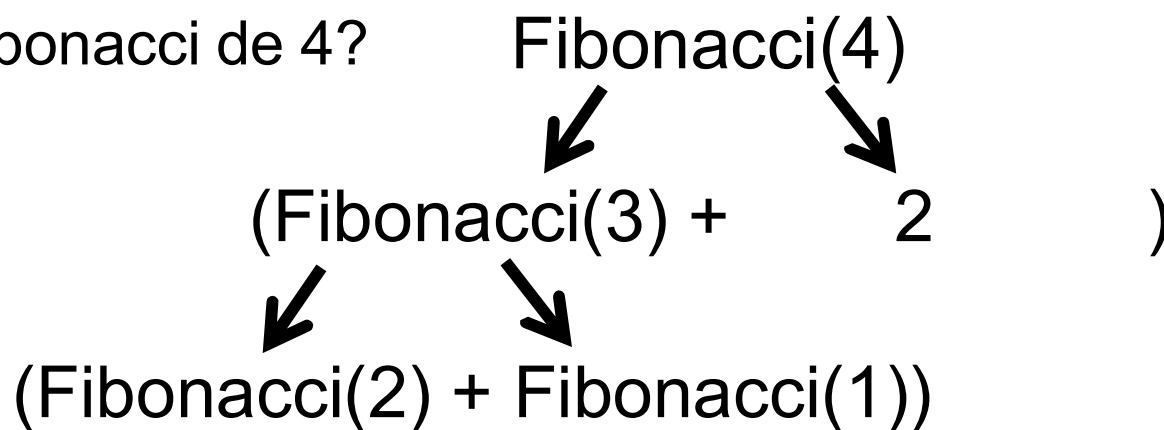
$$\begin{array}{c} \text{Fibonacci}(4) \\ \swarrow \quad \searrow \\ (\text{Fibonacci}(3) + 2) \end{array}$$

Exemplo: Fibonacci Recursivo

- Definição do Fibonacci é recursiva:

$$\left\{ \begin{array}{l} \text{Fib}(0) = 1 \\ \text{Fib}(1) = 1 \\ \text{Fib}(n) = \text{Fib}(n-1) + \text{Fib}(n-2) \end{array} \right.$$

- Qual é o Fibonacci de 4?

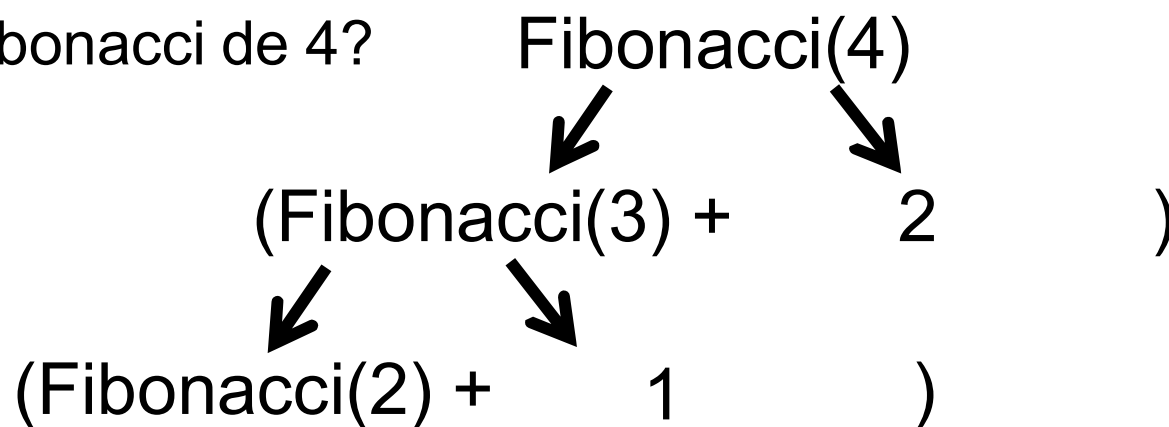


Exemplo: Fibonacci Recursivo

- Definição do Fibonacci é recursiva:

$$\left\{ \begin{array}{l} \text{Fib}(0) = 1 \\ \text{Fib}(1) = 1 \\ \text{Fib}(n) = \text{Fib}(n-1) + \text{Fib}(n-2) \end{array} \right.$$

- Qual é o Fibonacci de 4?

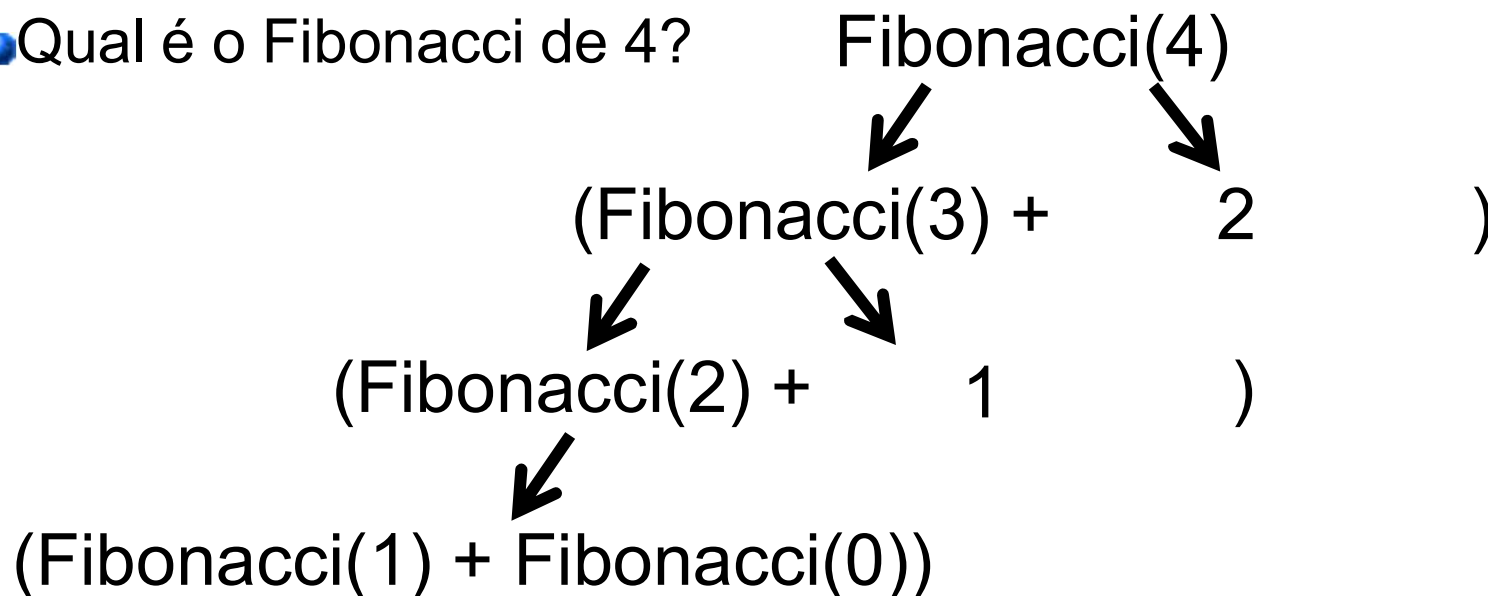


Exemplo: Fibonacci Recursivo

- Definição do Fibonacci é recursiva:

$$\left\{ \begin{array}{l} \text{Fib}(0) = 1 \\ \text{Fib}(1) = 1 \\ \text{Fib}(n) = \text{Fib}(n-1) + \text{Fib}(n-2) \end{array} \right.$$

- Qual é o Fibonacci de 4?

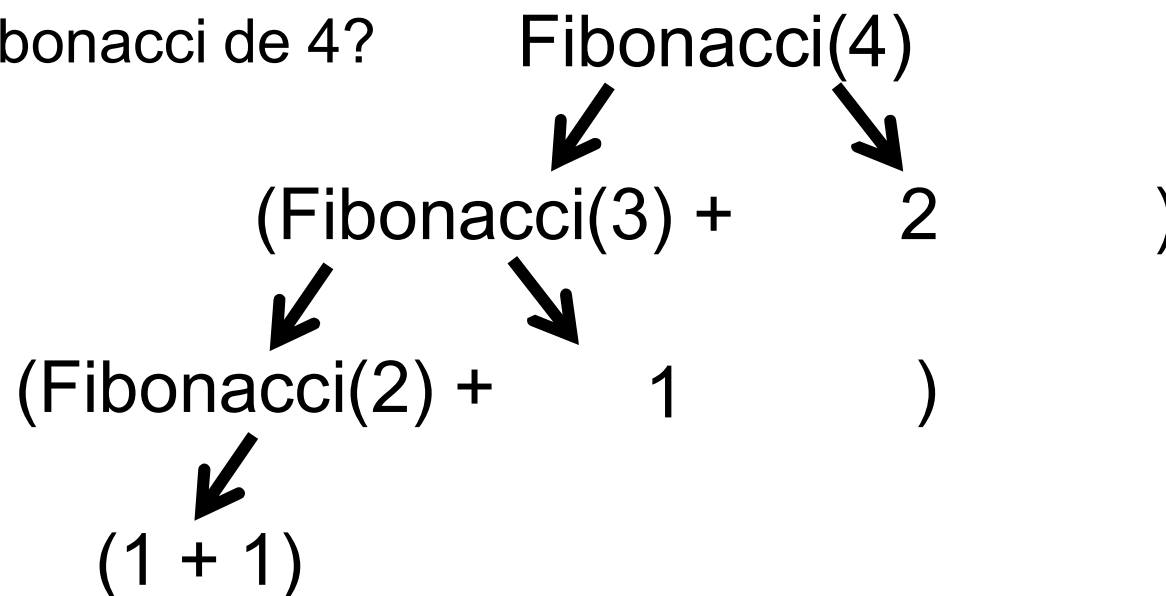


Exemplo: Fibonacci Recursivo

- Definição do Fibonacci é recursiva:

$$\left\{ \begin{array}{l} \text{Fib}(0) = 1 \\ \text{Fib}(1) = 1 \\ \text{Fib}(n) = \text{Fib}(n-1) + \text{Fib}(n-2) \end{array} \right.$$

- Qual é o Fibonacci de 4?

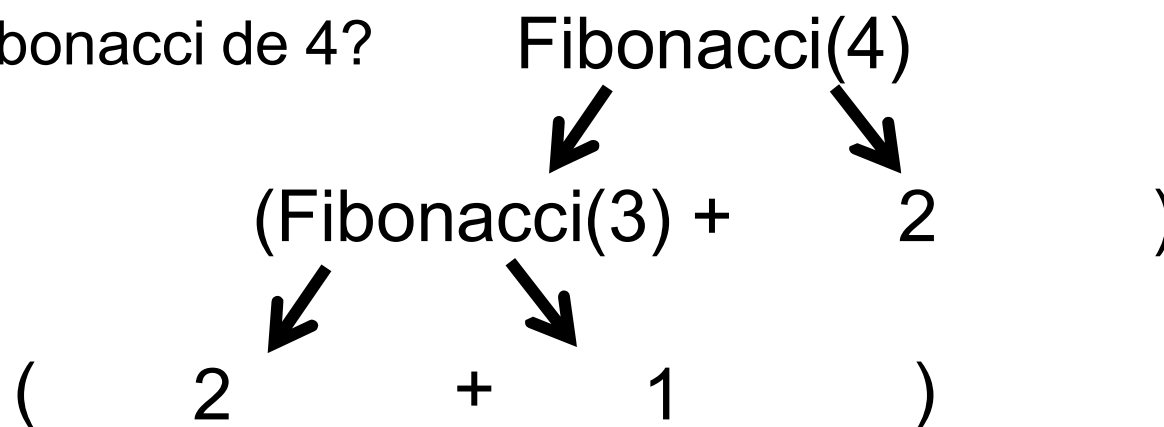


Exemplo: Fibonacci Recursivo

- Definição do Fibonacci é recursiva:

$$\left\{ \begin{array}{l} \text{Fib}(0) = 1 \\ \text{Fib}(1) = 1 \\ \text{Fib}(n) = \text{Fib}(n-1) + \text{Fib}(n-2) \end{array} \right.$$

- Qual é o Fibonacci de 4?



Exemplo: Fibonacci Recursivo

- Definição do fatorial é recursiva:

$$\left\{ \begin{array}{l} \text{Fib}(0) = 1 \\ \text{Fib}(1) = 1 \\ \text{Fib}(n) = \text{Fib}(n-1) + \text{Fib}(n-2) \end{array} \right.$$

- Qual é o Fibonacci de 4?

Fibonacci(4)

(3 + 2)

Exemplo: Fibonacci Recursivo

- Definição do fatorial é recursiva:

$$\left\{ \begin{array}{l} \text{Fib}(0) = 1 \\ \text{Fib}(1) = 1 \\ \text{Fib}(n) = \text{Fib}(n-1) + \text{Fib}(n-2) \end{array} \right.$$

- Qual é o Fibonacci de 4?

5

Exemplo: Fibonacci Recursivo

```
int fibonacci (int n){  
    int resp;  
    if (n == 0 || n == 1){  
        resp = 1;  
    } else {  
        resp = fibonacci(n - 1) + fibonacci(n - 2);  
    }  
    return resp;  
}  
void main(){  
    int valor = fibonacci(4);  
    print(valor);  
}
```


- Faça um método recursivo que receba dois números inteiros e retorne a multiplicação do primeiro pelo segundo fazendo somas

$$5 \times 13 = \underbrace{5 + 5 + \dots + 5}_{13 \text{ vezes}}$$

- Faça um método recursivo que receba dois números inteiros e retorne a multiplicação do primeiro pelo segundo fazendo somas

```
int multiplicacao (int a, int b){  
    int resp = 0;  
  
    if (b > 0){  
        resp = a + multiplicacao(a, b - 1);  
    }  
  
    return resp;  
}  
void main (...){  
    multiplicacao(4, 3);  
}
```

- Faça um método recursivo que receba dois números inteiros e retorne a multiplicação do primeiro pelo segundo fazendo somas (**outra resposta**)

```
int multiplicacao (int a, int b, int i){  
    int resp = 0;  
    if (i < b){  
        resp = a + multiplicacao(a, b, i + 1);  
    }  
    return resp;  
}  
  
int multiplicacao (int a, int b){  
    return multiplicacao(a, b, 0);  
}  
  
void main (...){  
    multiplicacao(4, 3);  
}
```

- Faça um método recursivo que receba dois números inteiros e retorne a multiplicação do primeiro pelo segundo fazendo somas **(outra resposta)**

```
int multiplicacao (int a, int b, int i){  
    int resp = 0;  
  
    if (i < b){  
        resp = a + multiplicacao(a, b, i + 1);  
    }  
  
    return resp;  
}  
  
int multiplicacao (int a, int b){  
    return multiplicacao(a, b, 0);  
}
```

```
int multiplicacao (int a, int b){  
    int resp = 0;  
  
    for (int i = 0; i < b; i = i + 1){  
        resp += a;  
    }  
  
    return resp;  
}
```

Exercício

- Faça um método recursivo que receba dois números inteiros e retorne a multiplicação do primeiro pelo segundo fazendo somas **(outra resposta)**

```
int multiplicacao (int a, int b, int i){  
    int resp = 0;  
  
    if (i < b){  
        resp = a + multiplicacao(a, b, i + 1);  
    }  
  
    return resp;  
}  
  
int multiplicacao (int a, int b){  
    return multiplicacao(a, b, 0);  
}
```

```
int multiplicacao (int a, int b){  
    int resp = 0;  
  
    for (int i = 0; i < b; i = i + 1){  
        resp += a;  
    }  
  
    return resp;  
}
```

Exercício

- Faça um método recursivo que receba dois números inteiros e retorne a multiplicação do primeiro pelo segundo fazendo somas (**outra resposta**)

```
int multiplicacao (int a, int b, int i){  
    int resp = 0;  
  
    if (i < b){  
        resp = a + multiplicacao(a, b, i + 1);  
    }  
  
    return resp;  
}  
  
int multiplicacao (int a, int b){  
    return multiplicacao(a, b, 0);  
}
```

```
int multiplicacao (int a, int b){  
    int resp = 0;  
  
    for (int i = 0; i < b; i = i + 1){  
        resp += a;  
    }  
  
    return resp;  
}
```

Exercício

- Faça um método recursivo que receba dois números inteiros e retorne a multiplicação do primeiro pelo segundo fazendo somas (**outra resposta**)

```
int multiplicacao (int a, int b, int i){  
    int resp = 0;  
  
    if (i < b){  
        resp = a + multiplicacao(a, b, i + 1);  
    }  
  
    return resp;  
}  
  
int multiplicacao (int a, int b){  
    return multiplicacao(a, b, 0);  
}
```

```
int multiplicacao (int a, int b){  
    int resp = 0;  
  
    for (int i = 0; i < b; i = i + 1){  
        resp += a;  
    }  
  
    return resp;  
}
```

- Faça um método recursivo que receba dois números inteiros e retorne a multiplicação do primeiro pelo segundo fazendo somas (**outra resposta**)

```
int multiplicacao (int a, int b, int i){  
    int resp = 0;  
  
    if (i < b){  
        multiplicacao(a, b, i + 1);  
    }  
  
    return resp;  
}  
  
int multiplicacao (int a, int b){  
    return multiplicacao(a, b, 0);  
}
```

```
int multiplicacao (int a, int b){  
    int resp = 0;  
  
    for (int i = 0; i < b; i = i + 1){  
    }  
  
    return resp;  
}
```


Exercício

- Faça um método recursivo que receba dois números inteiros e retorne a multiplicação do primeiro pelo segundo fazendo somas (**outra resposta**)

```
int multiplicacao (int a, int b, int i){  
    int resp = 0;  
  
    if (i < b){  
        resp = a + multiplicacao(a, b, i + 1);  
    }  
  
    return resp;  
}  
  
int multiplicacao (int a, int b){  
    return multiplicacao(a, b, 0);  
}
```

```
int multiplicacao (int a, int b){  
    int resp = 0;  
  
    for (int i = 0; i < b; i = i + 1){  
        resp += a;  
    }  
  
    return resp;  
}
```

- Faça um método recursivo para cada um dos problemas abaixo

$$\begin{cases} T(0) = 1 \\ T(1) = 2 \\ T(n) = T(n-1) * T(n-2) - T(n-1) \end{cases}$$

$$\begin{cases} T(0) = 1 \\ T(n) = T(n-1) + 2 \end{cases}$$

Considerações

- Todo programa iterativo pode ser feito de forma recursiva e vice-versa
 - Algumas vezes é mais “fácil” fazer um programa de forma recursiva!!!
- O conceito de recursividade é fundamental na computação e na matemática (por exemplo, número naturais, fatorial e outros)
- A recursividade pode ser direta ou indireta (A chama B que chama A)

Considerações

- O SO usa uma pilha para armazenar o estado corrente do programa antes de cada chamada não terminada e quando uma chamada termina, o SO recupera o estado armazenado na pilha
 - As variáveis locais são recriadas para cada chamada recursiva
- Por que na prática é importante manter um nível “limitado” de chamadas recursivas?

Considerações

- Um dos principais exemplos sobre recursividade apresentados na literatura (e neste material também!!!) é o Fibonacci
- Qual é a sua opinião sobre as implementações recursiva e iterativa do Fibonacci?

- Um procedimento que **chama a si** mesmo é dito ser **recursivo**.
- Recursividade permite descrever algoritmos de forma mais clara e concisa, especialmente **problemas recursivos** ou que utilizam **estruturas recursivas**

Implementação de Recursividade

- Usa-se uma **pilha** para armazenar os dados usados em cada chamada de um procedimento que ainda não terminou.
- Todos os dados não globais vão para a **pilha**, registrando o estado corrente da computação.
- Quando uma ativação anterior prossegue, os dados da pilha são recuperados.
- Uma definição recursiva consiste em duas partes:
 - 1 Caso base: elementos básicos
 - 2 Regras de construção de novos objetos a partir dos elementos básicos.

A **pilha** que armazena os procedimentos recursivos é implementado e gerenciado pelo compilador e não pelo desenvolvedor! Você só precisa ter o cuidado de não estourar sua capacidade!

```
1 int fatorial(int n) {  
2     // caso base  
3     if(n == 0)  
4         return 1;  
5     // regra de construcao  
6     return n * fatorial(n - 1);  
7 }
```



```
1 int fatorial(int n) {  
2     // caso base  
3     if(n == 0)  
4         return 1;  
5     // regra de construcao  
6     return n * fatorial(n - 1);  
7 }
```

Executando o procedimento recursivo:

fatorial(5) = 5 x fatorial(4)

Pilha
5 x fatorial(4)

```
1 int fatorial(int n) {  
2     // caso base  
3     if(n == 0)  
4         return 1;  
5     // regra de construcao  
6     return n * fatorial(n - 1);  
7 }
```

Executando o procedimento recursivo:

$\text{fatorial}(4) = 4 \times \text{fatorial}(3)$

Pilha
4 x fatorial(3)
5 x fatorial(4)

```
1 int fatorial(int n) {  
2     // caso base  
3     if(n == 0)  
4         return 1;  
5     // regra de construcao  
6     return n * fatorial(n - 1);  
7 }
```

Executando o procedimento recursivo:

fatorial(3) = 3 x fatorial(2)

Pilha
3 x fatorial(2)
4 x fatorial(3)
5 x fatorial(4)

```
1 int fatorial(int n) {  
2     // caso base  
3     if(n == 0)  
4         return 1;  
5     // regra de construcao  
6     return n * fatorial(n - 1);  
7 }
```

Executando o procedimento recursivo:

$\text{fatorial}(2) = 2 \times \text{fatorial}(1)$

Pilha
2 x fatorial(1)
3 x fatorial(2)
4 x fatorial(3)
5 x fatorial(4)

```
1 int fatorial(int n) {  
2     // caso base  
3     if(n == 0)  
4         return 1;  
5     // regra de construcao  
6     return n * fatorial(n - 1);  
7 }
```

Executando o procedimento recursivo:

fatorial(1) = 1 x fatorial(0)

Pilha
1 x fatorial(0)
2 x fatorial(1)
3 x fatorial(2)
4 x fatorial(3)
5 x fatorial(4)

```
1 int fatorial(int n) {  
2     // caso base  
3     if(n == 0)  
4         return 1;  
5     // regra de construcao  
6     return n * fatorial(n - 1);  
7 }
```

Executando o procedimento recursivo:

fatorial(0) = 1

Pilha
1 x fatorial(0)
2 x fatorial(1)
3 x fatorial(2)
4 x fatorial(3)
5 x fatorial(4)

```
1 int fatorial(int n) {  
2     // caso base  
3     if(n == 0)  
4         return 1;  
5     // regra de construcao  
6     return n * fatorial(n - 1);  
7 }
```

Executando o procedimento recursivo:

$\text{fatorial}(1) = 1 \times \text{fatorial}(0) = 1 \times 1 = 1$

Pilha
2 x fatorial(1)
3 x fatorial(2)
4 x fatorial(3)
5 x fatorial(4)

```
1 int fatorial(int n) {  
2     // caso base  
3     if(n == 0)  
4         return 1;  
5     // regra de construcao  
6     return n * fatorial(n - 1);  
7 }
```

Executando o procedimento recursivo:

$\text{fatorial}(2) = 2 \times \text{fatorial}(1) = 2 \times 1 = 2$

Pilha
3 x fatorial(2)
4 x fatorial(3)
5 x fatorial(4)


```
1 int fatorial(int n) {  
2     // caso base  
3     if(n == 0)  
4         return 1;  
5     // regra de construcao  
6     return n * fatorial(n - 1);  
7 }
```

Executando o procedimento recursivo:

$\text{fatorial}(3) = 3 \times \text{fatorial}(2) = 3 \times 2 = 6$

Pilha
4 x fatorial(3)
5 x fatorial(4)

```
1 int fatorial(int n) {  
2     // caso base  
3     if(n == 0)  
4         return 1;  
5     // regra de construcao  
6     return n * fatorial(n - 1);  
7 }
```

Executando o procedimento recursivo:

$\text{fatorial}(4) = 4 \times \text{fatorial}(3) = 4 \times 6 = 24$

Pilha
5 x fatorial(4)

```
1 int fatorial(int n) {  
2     // caso base  
3     if(n == 0)  
4         return 1;  
5     // regra de construcao  
6     return n * fatorial(n - 1);  
7 }
```

Executando o procedimento recursivo:

fatorial(5) = 5 x fatorial(4) = 5 x 24 = 120

Pilha

As Três Regras da Recursão

- 1 **Saber quando parar:** toda função recursiva deve fazer uma checagem para verificar se o problema foi completado antes da nova chamada recursiva.
- 2 **Decidir como aplicar o primeiro passo:** quebrar um problemas em subproblemas que possam ser resolvidos instantaneamente.
- 3 **Analisar o problema de forma que o mesmo possa ser dividido em problemas menores:** achar uma maneira da função chamar a si mesma recursivamente e cujo parâmetro é um problema menor resultante da segunda regra.

Quando Não Usar Recursividade

- Nem todo problema de natureza recursiva deve ser resolvido com um algoritmo recursivo.

Cálculo dos números de Fibonacci

$$f_0 = 0, f_1 = 1,$$

$$f_n = f_{n-1} + f_{n-2}, \text{ para } n \geq 2$$

Exemplo

Exemplo:

$$F = 1, 1, 2, 3, 5, 8, 13, 21, 34, 55, 89, 144, 233, 377, \dots$$

Quando Não Usar Recursividade

Procedimento Recursivo

```
1 int fibonacci(int n) {  
2     if (n < 2)  
3         return n;  
4     else  
5         return fibonacci(n-1) + fibonacci(n-2);  
6 }
```

- É extremamente ineficiente porque recalcula o mesmo valor várias vezes.

Quando Não Usar Recursividade

Versão iterativa

```

1  int fibonacci_ite(int n) {
2      int F=0, k=1 ;
3      for(int i=1; i<=n; i++) {
4          F += k;
5          k = F - k;
6      }
7      return F;
8  }
```

- O programa tem complexidade de tempo $O(n)$ e de espaço $O(1)$.
- Evitar uso de recursividade quando existe solução óbvia por iteração.
- Comparação versões recursiva e iterativa:

n	20	30	40	50
Recursiva	1 seg	2 min	21 dias	10^9 anos
Iterativa	1/3 mseg	1/2 mseg	3/4 mseg	1,5 mseg

Exercício

- Faça uma função que recebe como parâmetro um inteiro positivo N e retorne o somatório de todos os números inteiros entre 0 e N .

Exercício - Solução

- Faça uma função recursiva que recebe como parâmetro um inteiro positivo N e retorne o somatório de todos os números inteiros entre 0 e N .

```
1 int somatorio(int n) {  
2     if (n == 1)  
3         return 1;  
4     else  
5         return n + somatorio(n-1);  
6 }
```

Exercício

- Escreva um procedimento recursivo para converter um número decimal para a forma binária.

Dica

Uma maneira simples de resolver o problema é dividir o número decimal sucessivamente por 2 e pegar o resto da i -ésima divisão, da direita para a esquerda.

Por exemplo, para o número 12 temos:

$$12/2 = 6, \text{ resto } 0$$

$$6/2 = 3, \text{ resto } 0$$

$$3/2 = 1, \text{ resto } 1$$

$$1/2 = 0, \text{ resto } 1$$

Portanto, o número 12 em binário é 1100.

Exercício - Solução

- Escreva um procedimento recursivo para converter um número decimal para a forma binária.

```
1 void dec2bin(int n) {  
2     if (n <= 0)  
3         return;  
4     else {  
5         dec2bin(n/2);  
6         print (n%2);  
7     }  
8 }
```

Próxima aula

- Noções de Complexidade