



ALGORITMOS

PROF^ª: SIMONE DOMINICO

Laços de repetição



Algoritmos

Condicionais



Laços de repetição



OBJETIVO

LAÇOS DE REPETIÇÃO



LAÇOS DE REPETIÇÃO

- Repetições com **Enquanto** (while) são adequados em situações nas quais o número de iterações é desconhecido
- O comando **Para** (for) ao contrário, tem uso indicado nas situações em que o número de iterações é conhecido a priori

LAÇOS DE REPETIÇÃO — ENQUANTO

Escreva um algoritmo para calcular o fatorial de um número informado pelo usuário

Entradas: um número.

Saída: fatorial do número.

Processamento: realizar operações necessárias para calcular o fatorial.

ANALISANDO — FATORIAL

1. O fatorial de um número n é dado pela seguinte equação matemática:

$$f = n \times (n - 1) \times (n - 2) \times \dots \times 1$$

- A quantidade de termos da equação depende do valor de n

ANALISANDO — FATORIAL

1. O fatorial de um número n é dado pela seguinte equação matemática:

$$f = n \times (n - 1) \times (n - 2) \times \dots \times 1$$

- A quantidade de termos da equação depende do valor de n

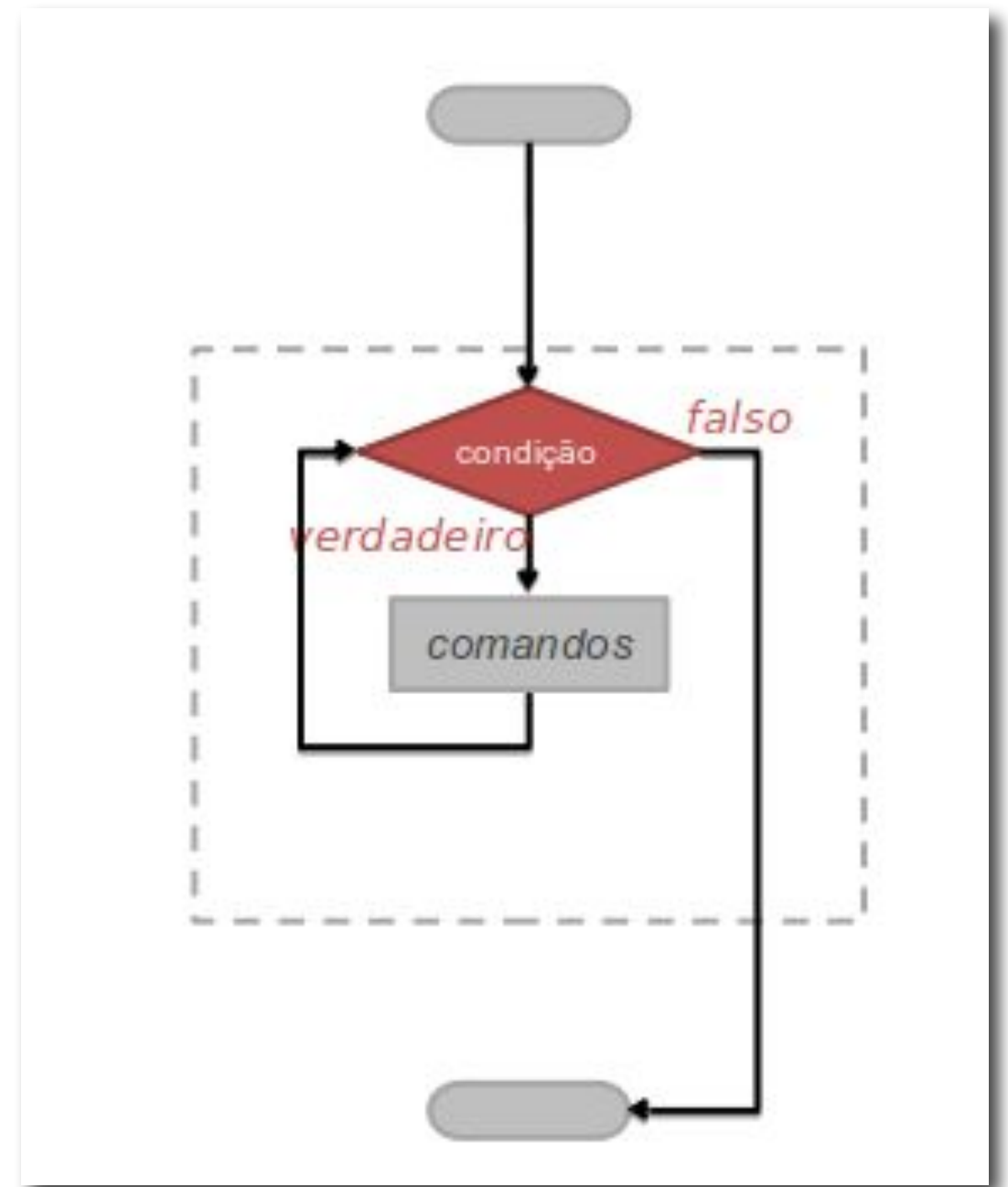
Exemplo: o fatorial de 4 é dado pelo que segue

$$4! = 4 \times 3 \times 2 \times 1 = 24$$

$$\text{Fatoriais} \left\{ \begin{array}{l} 0! = 1 \\ 1! = 1 \\ 2! = 2 \\ 3! = 6 \\ 4! = 24 \end{array} \right.$$

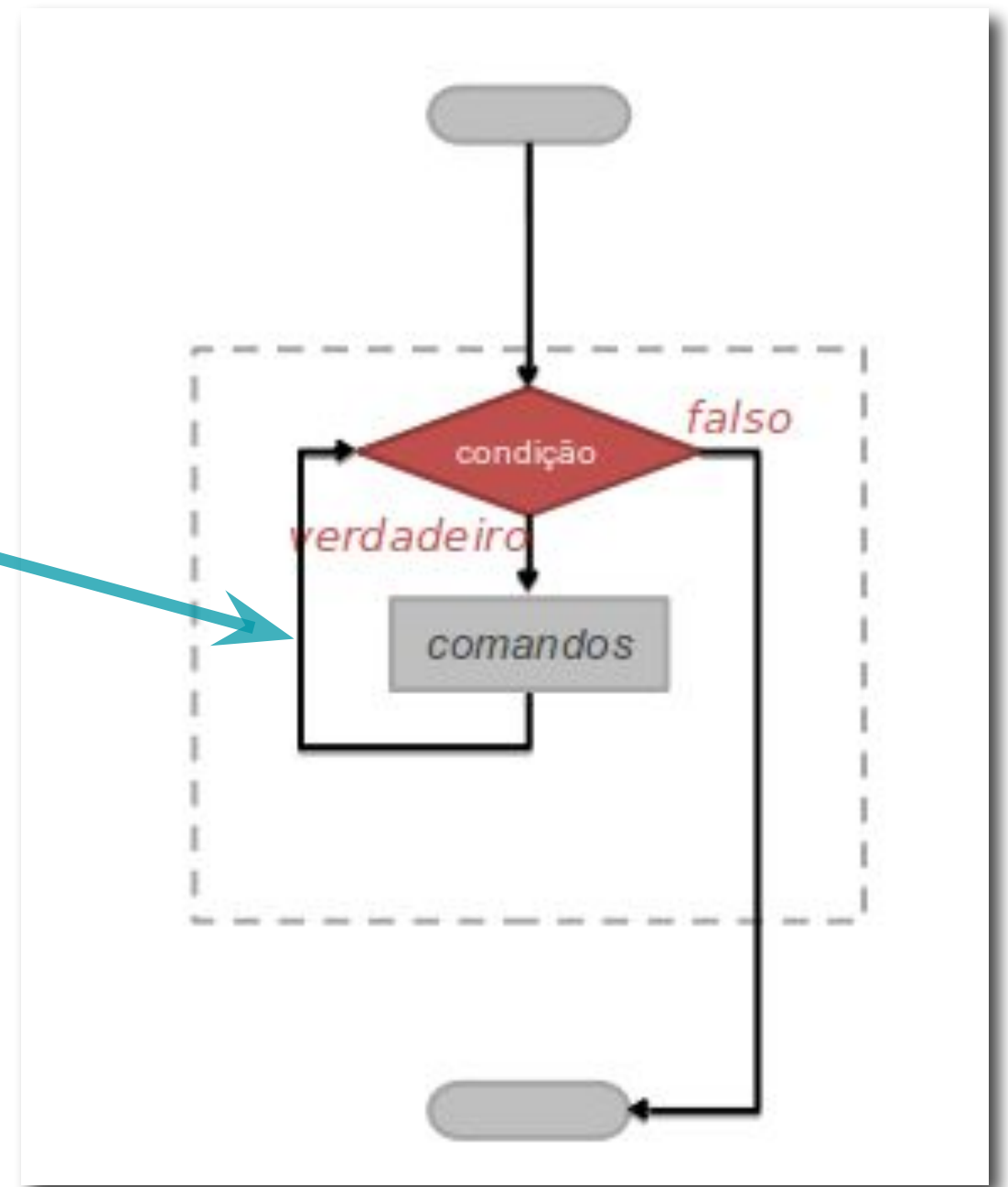
LAÇOS DE REPETIÇÃO

1. Uma repetição executa várias vezes os comandos no caminho verdadeiro
2. A quantidade de execuções depende de uma condição de parada
3. Quando condição for falsa o algoritmo sai do laço.



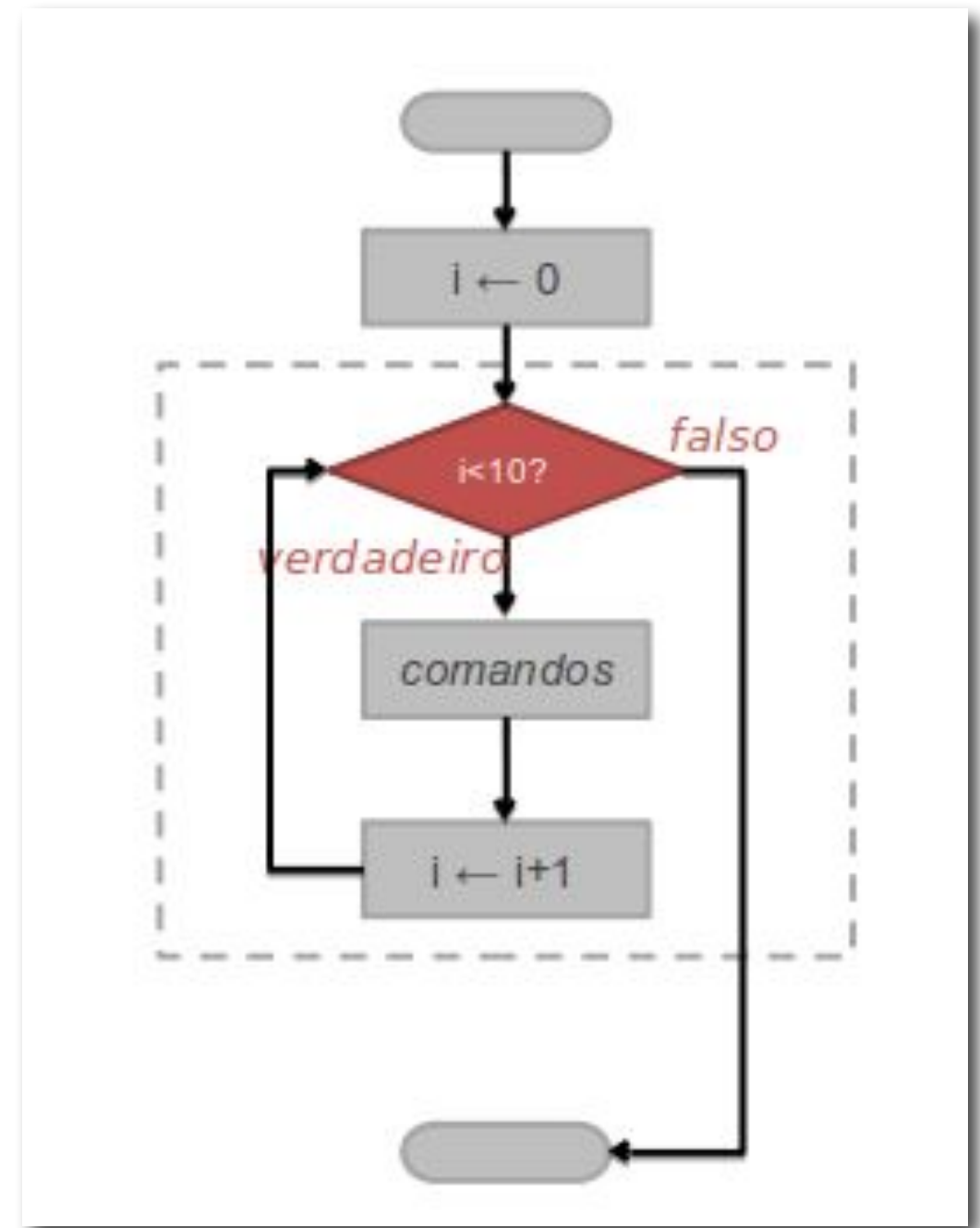
LAÇOS DE REPETIÇÃO

1. O que indica que existe uma repetição é a seta de volta para um ponto anterior.



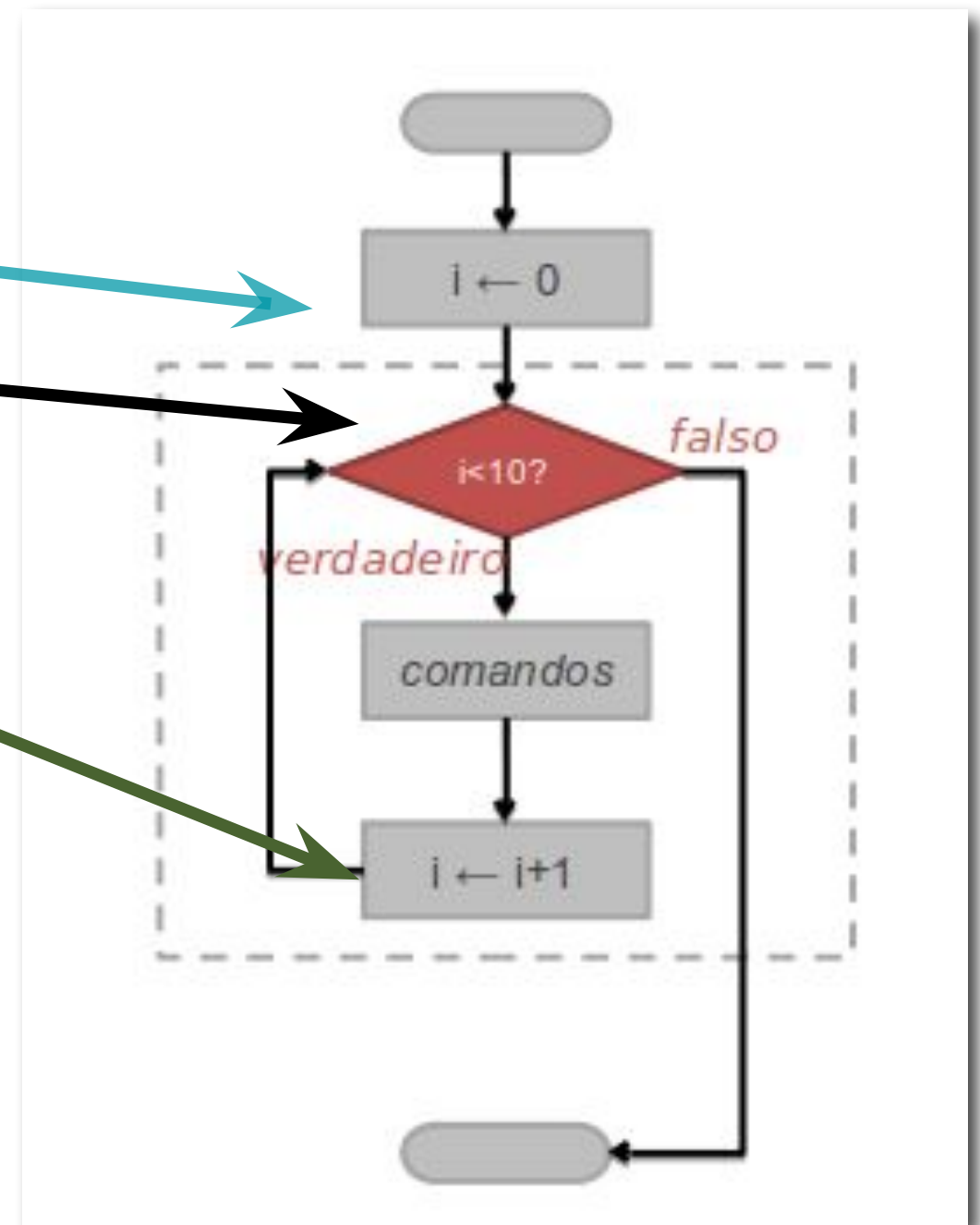
LAÇOS DE REPETIÇÃO

1. Exemplo: executa 10 vezes uma série de comandos.



LAÇOS DE REPETIÇÃO

1. Variável de controle
 1. Nesse caso usamos i
 1. Inicialização
 2. Condição
 3. Alteração
3. Condição de parada
 2. Algum dos comandos no caminho verdadeiro precisa alterar a variável de controle para que o laço não execute infinitamente.

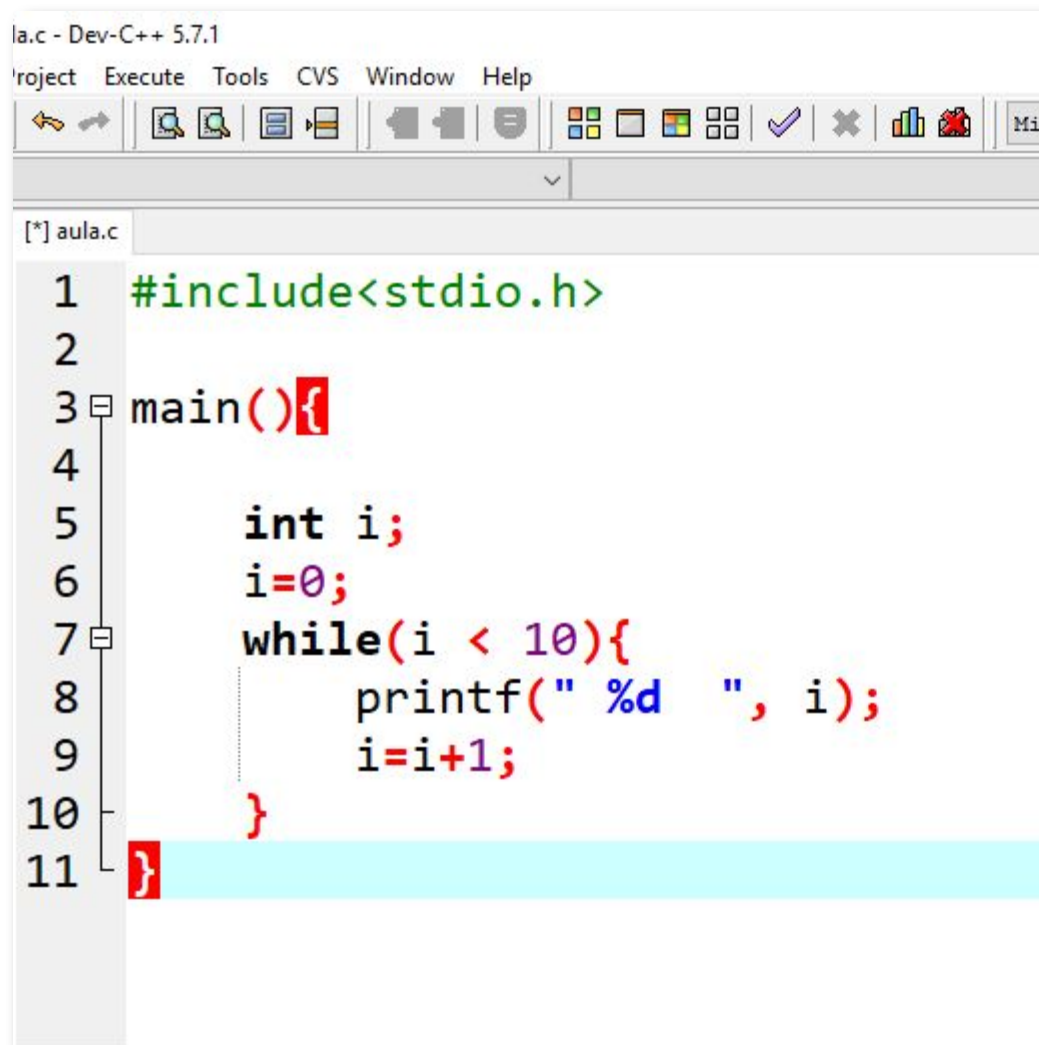


REPETIÇÃO EM PSEUDOCÓDIGO COM ENQUANTO

```
ALGORITMO CONTADOR
INTEIRO I
INICIO
    I=0
    ENQUANTO(I<10) FAÇA
        ESCREVA(" ", I, " ")
        I = I + 1
    FIMENQUANTO
FIM
```

1. Repete os comandos enquanto a condição for verdadeira.
2. A condição pode ser qualquer condição lógica
3. $i < 10$ é apenas um exemplo.

REPETIÇÃO EM C COM ENQUANTO (WHILE)



```
la.c - Dev-C++ 5.7.1
Project Execute Tools CVS Window Help

[*] aula.c
1  #include<stdio.h>
2
3  main(){
4
5      int i;
6      i=0;
7      while(i < 10){
8          printf(" %d ", i);
9          i=i+1;
10     }
11 }
```

1. Repete os comandos enquanto a condição for verdadeira.
2. A condição pode ser qualquer condição lógica
3. $i < 10$ é apenas um exemplo.

VOLTANDO AO CÁLCULO O FATORIAL

1. Tente implementar o exemplo do fatorial em pseudo-código.

OUTRO EXEMPLO DE PROBLEMA COM REPETIÇÃO

1. Faça um algoritmo que receba um valor e imprima a sua raiz quadrada
 1. O programa deve aceitar apenas valores de entrada positivos
 2. Caso o usuário digite um valor negativo deve ser solicitado outro valor

Como implementar essa consistência?

OUTRO EXEMPLO DE PROBLEMA COM REPETIÇÃO

1. Condição de parada
valor ≥ 0
2. Variável de controle
valor
3. Note que, uma vez no laço, a variável valor precisa ser alterada para que o algoritmo saia da repetição
4. Nesse caso a variável de controle é alterada pelo usuário, não por um incremento.

EXEMPLOS COM PSEUDO-CÓDIGO QUAIS SÃO AS SAÍDAS GERADAS?

```
ALGORITMO CONTADOR
INTEIRO I
INICIO
    I=0
    ENQUANTO(I<10) FAÇA
        ESCREVA(" ",I," ")
        I = I + 1
    FIMENQUANTO
    ESCREVA(I)

FIM
```

```
[*] aula.c
1  #include<stdio.h>
2
3  main(){
4
5      int i;
6      i=0;
7      while(i < 10){
8          printf(" %d ", i);
9          i=i+1;
10     }
11     printf(" %d ", i);
12 }
```

EXEMPLOS COM PSEUDO-CÓDIGO QUANTAS VEZES OS LAÇOS REPETEM?

```
ALGORITMO CONTADOR
INTEIRO I
INICIO
    I=0
    ENQUANTO(I<=10) FAÇA
        ESCREVA(" ",I, " ")
        I = I + 2
    FIMENQUANTO
FIM
```

```
aula.c
1  #include<stdio.h>
2
3  main(){
4
5      int i;
6      i=0;
7      while(i <= 10){
8          printf(" %d ", i);
9          i=i+2;
10     }
11 }
```

EXEMPLOS COM PSEUDO-CÓDIGO QUANTAS VEZES OS LAÇOS REPETEM?

```
ALGORITMO CONTADOR
INTEIRO I
INICIO
    I=10
    ENQUANTO(I< 10) FAÇA
        ESCREVA(" ",I, " ")
        I = I + 2
    FIMENQUANTO
FIM
```

```
aula.c
1  #include<stdio.h>
2
3  main(){
4
5      int i;
6      i=10;
7      while(i < 10){
8          printf(" %d ", i);
9          i=i+2;
10     }
11 }
```

EXEMPLOS COM PSEUDO-CÓDIGO

QUANTAS VEZES OS LAÇOS REPETEM?

1. E esse laço?
2. Quantas vezes repete?

```
aula.c
1  #include<stdio.h>
2
3  main(){
4
5      int entrada;
6      entrada=1;
7      while(entrada==1){
8          printf(" Deseja continuar? \n 0:Nao \n 1:Sim\n");
9          scanf("%d",&entrada);
10     }
11 }
```


EXEMPLOS COM PSEUDO-CÓDIGO

QUANTAS VEZES OS LAÇOS REPETEM?

1. E esse laço?
2. Quantas vezes repete?

aula.c

```
1  #include<stdio.h>
2
3  main(){
4
5      int entrada;
6      entrada=1;
7      while(entrada==
8          printf(" De
9          scanf("%d",
10
11 }
```

Nem sempre se
sabe de antemão
quantas vezes
um laço vai
executar

```
\n 1:Sim\n");
```

LAÇOS DE REPETIÇÃO (FOR)

- Escreva um algoritmo que leia as notas (P1, P2 e P3) de uma turma de 20 alunos
- Calcule a média de cada um dos alunos mostrando para o usuário
 - Entradas: notas P1, P2 e P3 (20 vezes)
 - Saída: 20 médias
 - Processamento: solicitar as 3 notas para cada aluno calculando cada média individualmente

SOLUÇÃO COM ENQUANTO(WHILE)

ALGORITMO CONTADOR

INTEIRO I

REAL NOTA1,NOTA2,NOTA3,MEDIA

INICIO

I=1

ENQUANTO(I<= 20) FAÇA

 ESCREVA("DIGITE A NOTA 1, DO ALUNO",I," ")

 LEIA(NOTA1)

 ESCREVA("DIGITE A NOTA 2, DO ALUNO",I," ")

 LEIA(NOTA2)

 ESCREVA("DIGITE A NOTA 3, DO ALUNO",I," ")

 LEIA(NOTA3)

 MEDIA = (NOTA1+NOTA2+NOTA3)/3

 ESCREVA("A MÉDIA É: ", MEDIA)

 I++

FIMENQUANTO

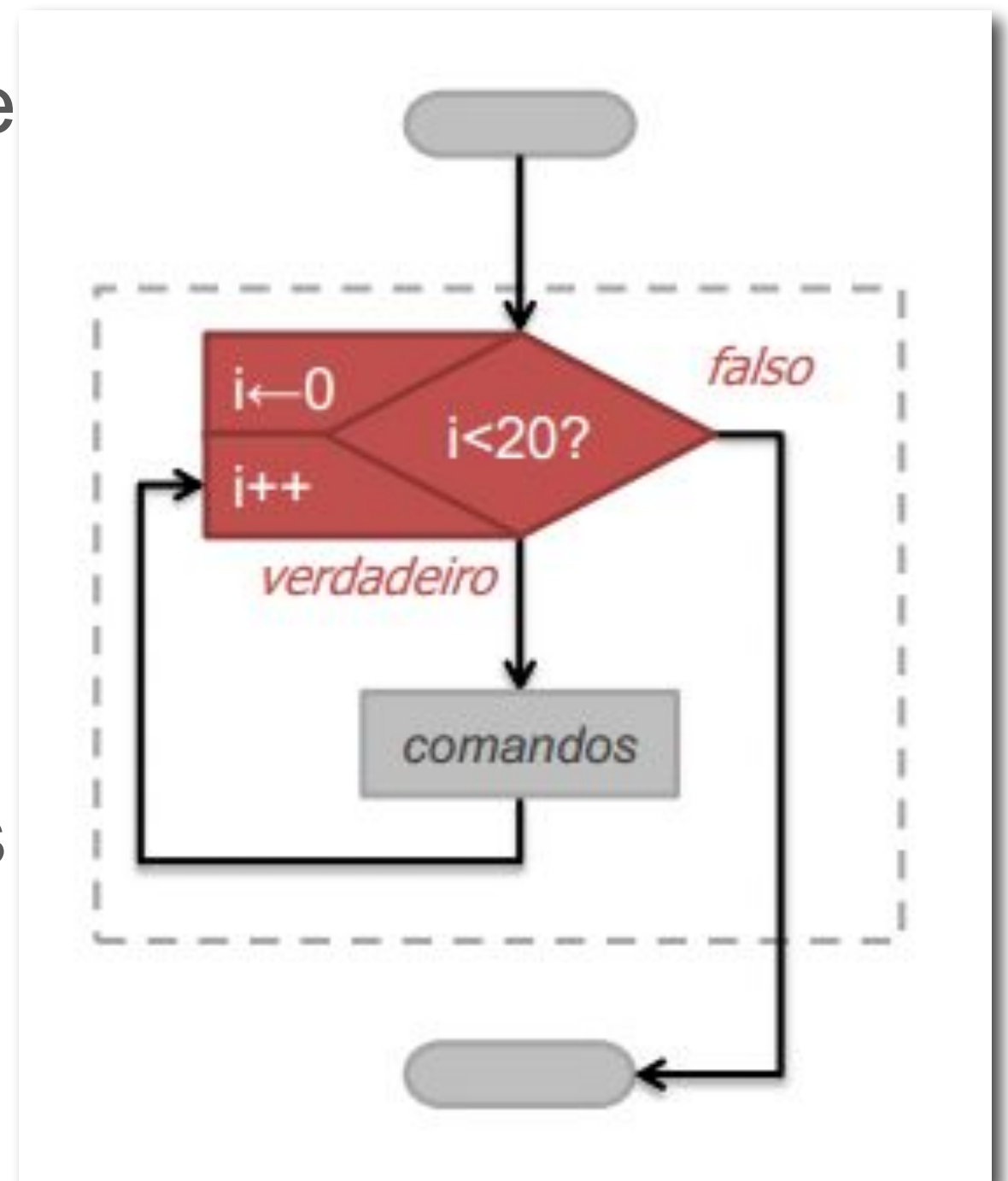
FIM

SOLUÇÃO COM ENQUANTO(WHILE)

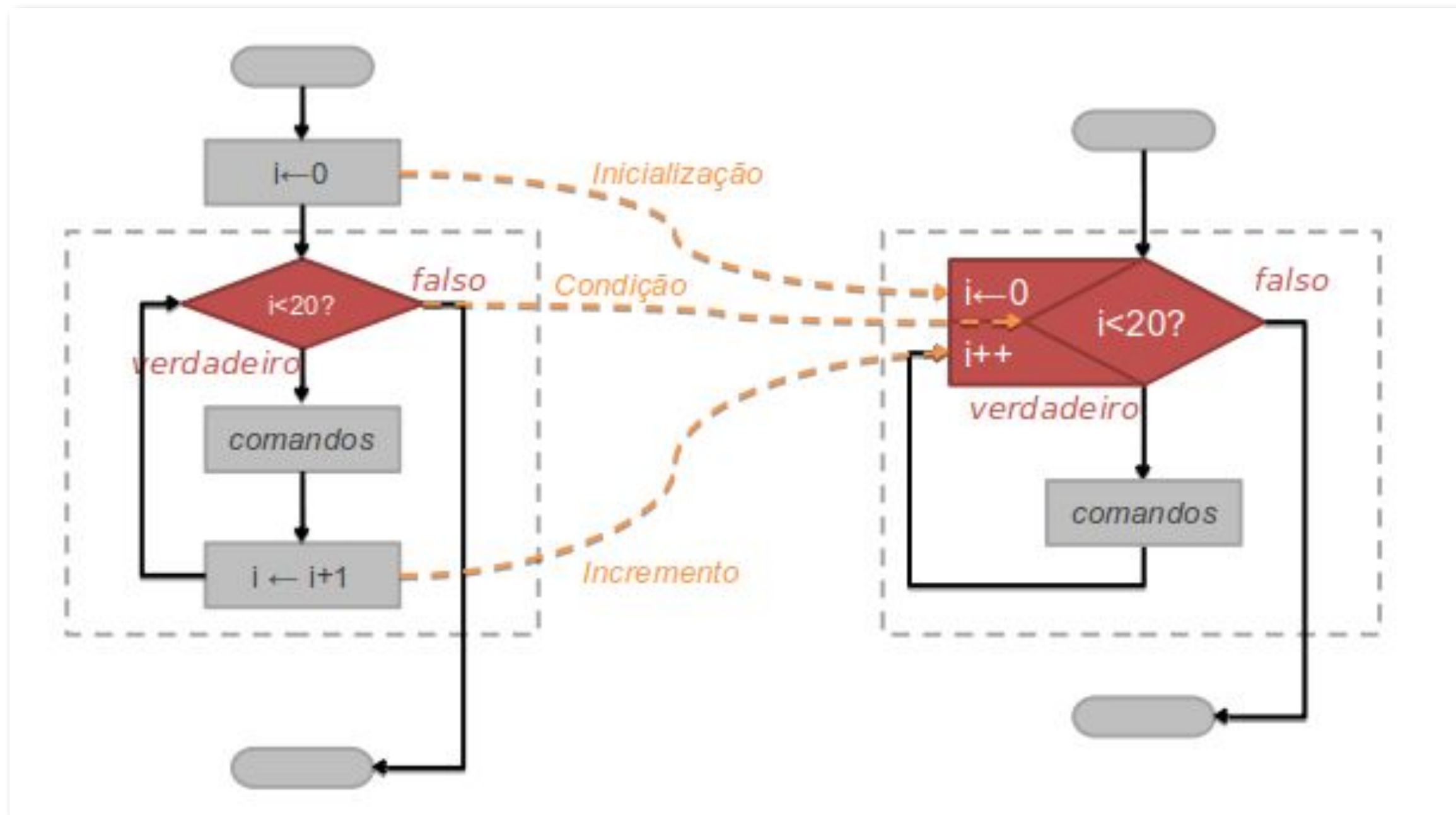
```
aula.c
1  #include<stdio.h>
2
3  main(){
4
5      float nota1,nota2,nota3, media;
6      int i=1;
7      while(i <= 20){
8          printf("Digite a nota 1 do aluno %d: ",i);
9          scanf("%f", &nota1);
10         printf("Digite a nota 2 do aluno %d: ",i);
11         scanf("%f", &nota2);
12         printf("Digite a nota 3 do aluno %d: ",i);
13         scanf("%f", &nota3);
14         media = (nota1+nota2+nota3)/3;
15         printf("A media eh: %.2f\n", media);
16         i++;
17     }
18 }
```

LAÇOS DE REPETIÇÃO - PARA (FOR)

- Inicialização, condição e incremento da variável de controle aparecem todos juntos.
- A grande virtude da repetição com Para é organizar todas as expressões necessárias em um lugar só.



COMPARANDO ENQUANTO (WHILE) E PARA(FOR)



PARA PSEUDOCÓDIGO

1. Início

2. Para ($i \leftarrow 0 ; i < 20 ; i++$)

3. Faça

4. Comandos ;

5. Fim

6. Fim

- Inicialização

– $i \leftarrow 0$

- Condição

– $i < 20$

- Incremento

– $i++$

Vai executar os comandos 20 vezes com i variando entre 0 e 19!

EM C

```
[*] aula.c
1  #include<stdio.h>
2
3  main(){
4
5      int i;
6      for(i=0;i<20;i++){
7          /*comandos*/
8      }
9  }
```

Vai executar os comandos 20 vezes com *i* variando entre 0 e 19!

RETOMANDO AS NOTAS

ALGORITMO CONTADOR

INTEIRO I

REAL NOTA1,NOTA2,NOTA3,MEDIA

INICIO

PARA(I=1;I<= 20;I++) FAÇA

ESCREVA("DIGITE A NOTA 1, DO ALUNO",I," ")

LEIA(NOTA1)

ESCREVA("DIGITE A NOTA 2, DO ALUNO",I," ")

LEIA(NOTA2)

ESCREVA("DIGITE A NOTA 3, DO ALUNO",I," ")

LEIA(NOTA3)

MEDIA = (NOTA1+NOTA2+NOTA3)/3

ESCREVA("A MÉDIA É: ", MEDIA)

FIMPARA

FIM

RETOMANDO AS NOTAS

```
aula.c
1  #include<stdio.h>
2
3  main(){
4
5      float nota1,nota2,nota3, media;
6      int i;
7      for(i=1;i<=20;i++){
8          printf("Digite a nota 1 do aluno %d: ",i);
9          scanf("%f", &nota1);
10         printf("Digite a nota 2 do aluno %d: ",i);
11         scanf("%f", &nota2);
12         printf("Digite a nota 3 do aluno %d: ",i);
13         scanf("%f", &nota3);
14         media = (nota1+nota2+nota3)/3;
15         printf("A media eh: %.2f\n", media);
16     }
17 }
```


Faca...enquanto

- Escreva um algoritmo que calcule a sequência de Fibonacci até o primeiro termo que ultrapasse o valor 10000
 - Entradas: nenhuma
 - Saída: lista de termos da série
 - Processamento: calcular cada termo da sequência de Fibonacci até encontrar um termo maior que 10000

Faca...enquanto

- É uma sequência de valores naturais, onde os dois primeiros termos são 0 e 1, e cada termo subsequente corresponde a soma dos dois anteriores

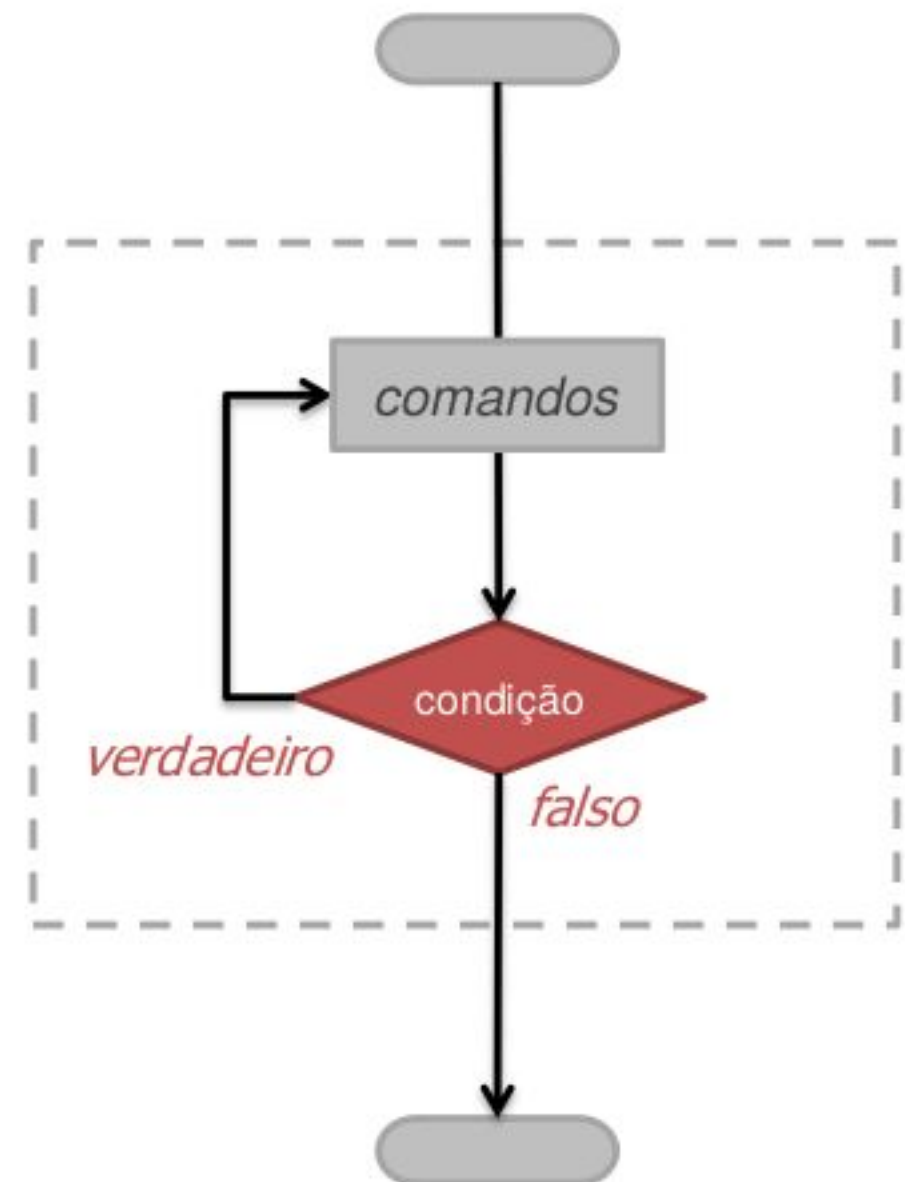
Primeiros 13 termos:

1º – 0	9º – 21
2º – 1	10º – 34
3º – 1	11º – 55
4º – 2	12º – 89
5º – 3	13º – 144
6º – 5	...
7º – 8	
8º – 13	

Existe uma certa discussão se o zero faz parte da sequência ou não, vamos assumir aqui que sim!

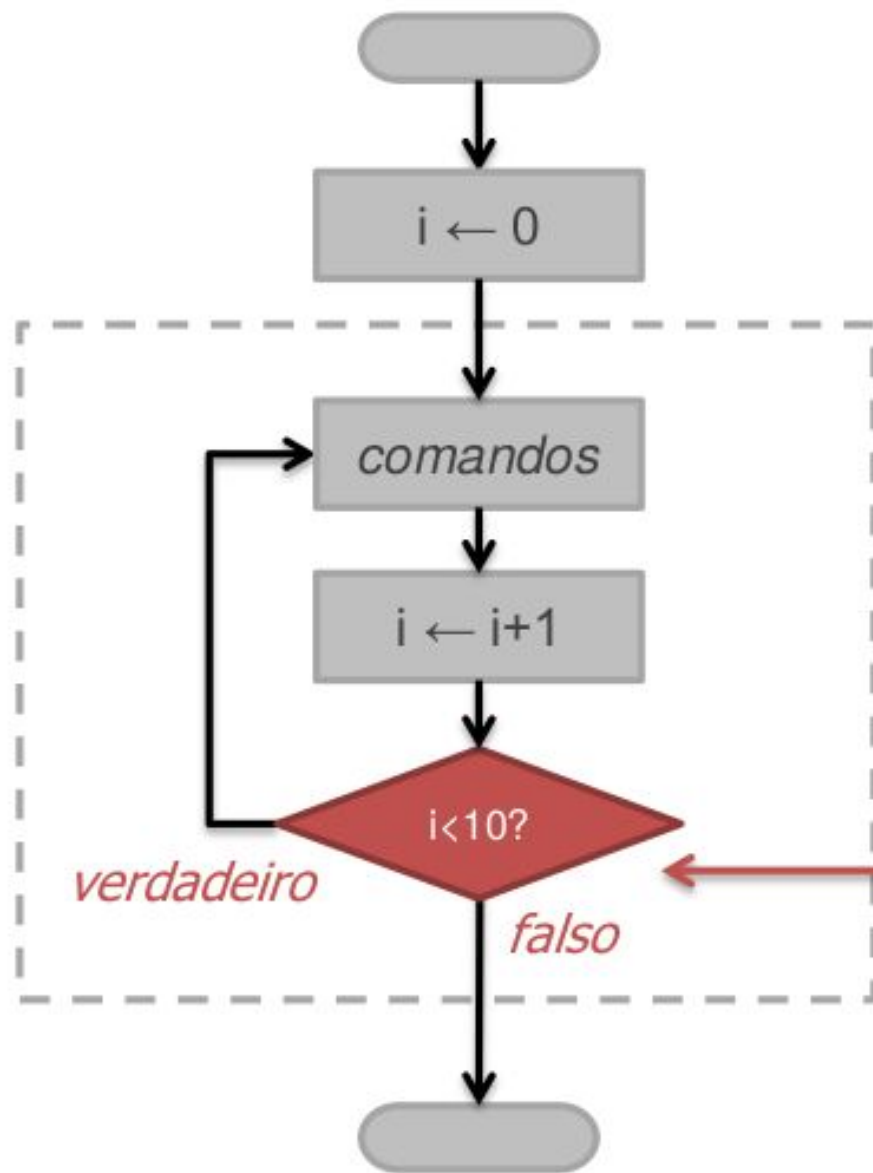
Faca...enquanto

- Executa normalmente um conjunto de comandos até chegar a condição
- Caso a condição resulte verdadeiro a execução retorna a um comando anterior do fluxograma
- Quando condição for falsa o algoritmo sai do laço e segue a execução normal



Faça...enquanto vs Enquanto

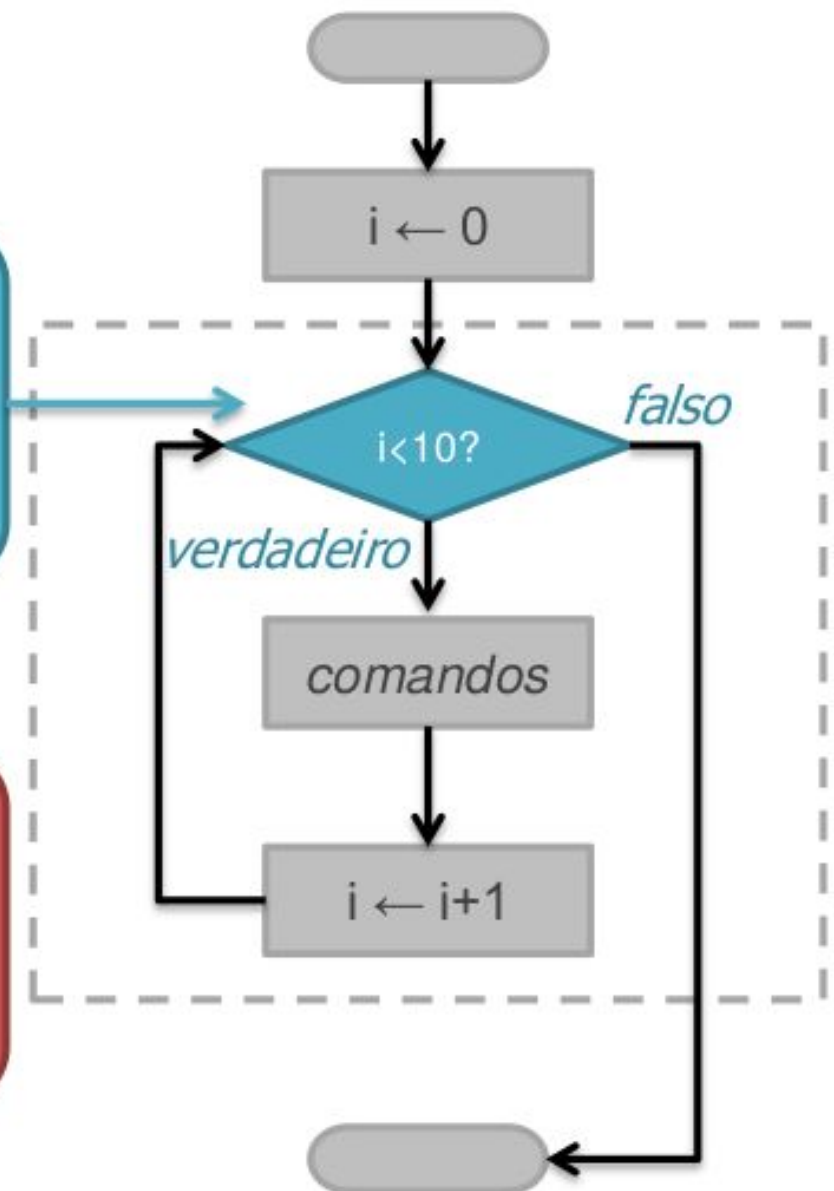
Faça-Enquanto



Testa a condição primeiro e depois executa os comandos

Executa os comandos e depois testa a condição

Enquanto



Faca...enquanto

```
ALGORITMO EXEMPLO
INTEIRO I

INICIO
  I=0
  FACA
    ESCREVA("OLÁ")
    I++;
  ENQUANTO(I<10)
FIM
```

Repete os
comandos
enquanto a
condição for
verdadeira

Faca...enquanto

Faça...enquanto

Faça-Enquanto

ALGORITMO EXEMPLO

INTEIRO I

INICIO

I=0

FACA

ESCREVA("OLÁ")

I++;

ENQUANTO(I<10)

FIM

Enquanto

ALGORITMO EXEMPLO

INTEIRO I

INICIO

I=0

ENQUANTO(I<10)FACA

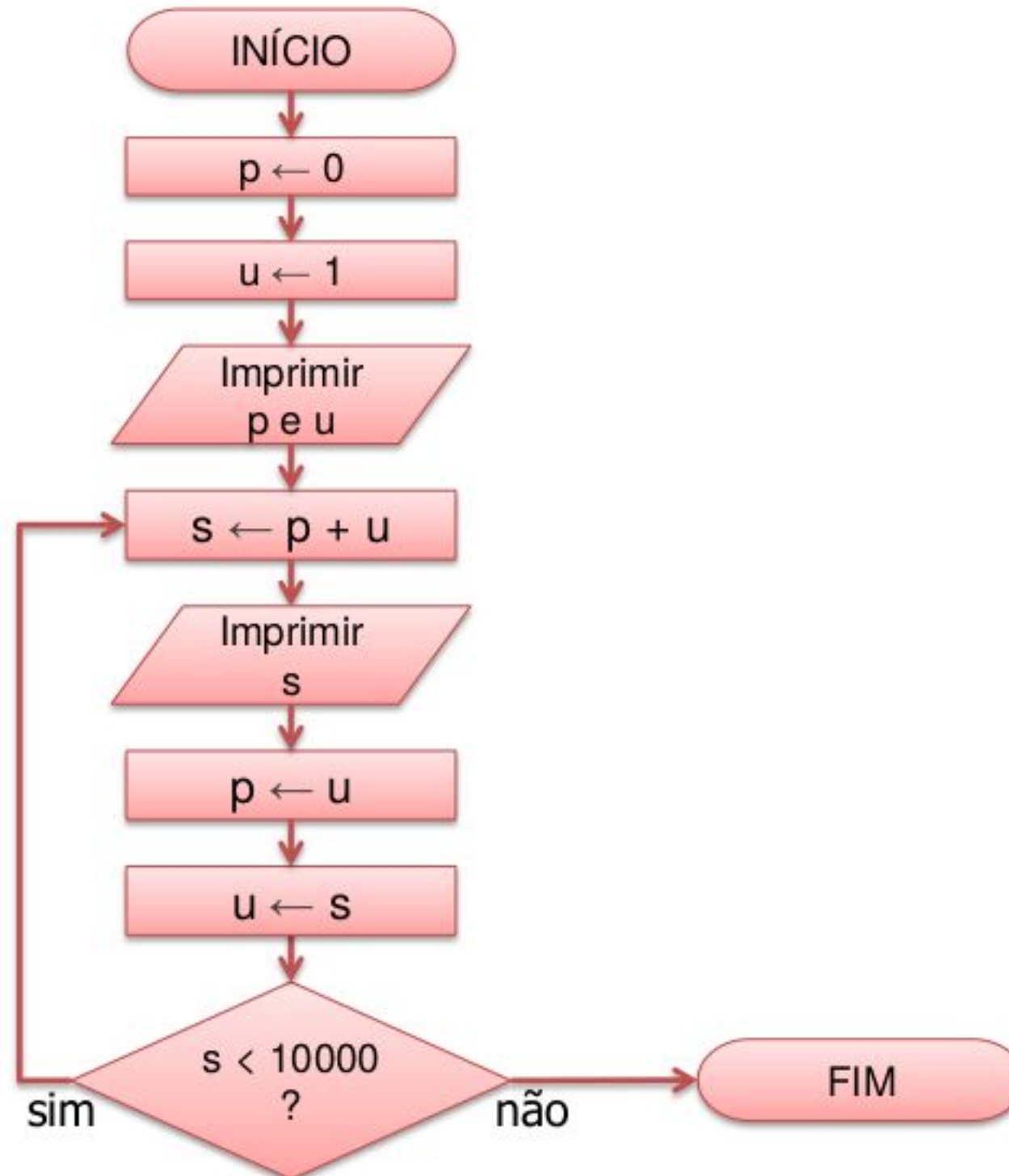
ESCREVA("OLÁ")

I++;

FIMENQUANTO

FIM

Faca...enquanto



RESUMINDO

- Repetições permitem que se execute um conjunto de comandos várias vezes.
- A variável ou variáveis que fazem parte da condição de parada deverão ter seu valor atribuído através de um comando de leitura, atribuição ou incremento ao longo da repetição.
- Quando se sabe exatamente quantas repetições serão necessárias, em geral, se aplica o comando Para (for).
- Quanto essa quantidade de repetições é incerta, o ideal é utilizar Enquanto (while).
- Quando precisa entrar pelo menos uma vez no laço usamos faça...enquanto (do...while)

TESTE DE MESA

- Um número inteiro x é perfeito se a soma de seus fatores (divisores), exceto ele mesmo, é igual a x . Por exemplo, 6 é perfeito visto que $1 + 2 + 3 = 6$. Escreva um programa para informar se x é um número perfeito.
- SAÍDA: mensagem — É PERFEITO ou NÃO É PERFEITO
- ENTRADA: UM NÚMERO (x)
- PROCESSAMENTO: como transformar os insumos na saída?

TESTE DE MESA

- SAÍDA: x “é um número perfeito” ou x “não é um número perfeito”
- ENTRADA: o número x
- PROCESSAMENTO: Somar (acumular) todos os divisores de X

Se (somaDosDivisores = X) então

Informe: X é um numero perfeito

Senão ...

Informe: X não é um numero perfeito

TESTE DE MESA

ALGORITMO CONTADOR

INTEIRO X,Y,RESTO,SOMA=0,METADE

INICIO

ESCREVA("DIGITE O VALOR QUE DESEJA TESTAR: ")

LEIA(X)

METADE = $X/2$

PARA(Y=1;Y<=METADE;Y++)FACA

RESTO = $X \% Y$

SE(RESTO==0)ENTAO

SOMA = SOMA + Y

FIMSE

FIMPARA

SE(SOMA==X)ENTAO

ESCREVA(X, " É UM NÚMERO PERFEITO")

SENAO

ESCREVA(X, " NÃO É UM NÚMERO PERFEITO")

FIMSE

FIM

TESTE DE MESA

```
[*] aula.c
1  #include<stdio.h>
2  main(){
3
4      int x,y,resto,soma,metade;
5      soma = 0;
6      printf("Digite o número que deseja testar: ");
7      scanf("%d", &x);
8      metade = x/2;
9      for(y=1;y<=metade;y++){
10         resto = x % y;
11         if(resto==0){
12             soma = soma + y;
13         }
14     }
15     if(soma==x){
16         printf("O numero %d eh perfeito",x);
17     }else{
18         printf("O numero %d nao eh perfeito",x);
19     }
20 }
```

TESTE DE MESA

- Em cada etapa acompanhamos o valor das variáveis envolvidas

LAÇOS DE REPETIÇÃO

1. Escreva um algoritmo que calcule a média dos números digitados pelo usuário, se eles forem pares. Termine a leitura se o usuário digitar zero (0).
2. Escreva um algoritmo que leia valores inteiros e encontre o maior e o menor deles. Termine a leitura se o usuário digitar zero (0);
3. Crie um algoritmo que conte até 100.
4. Escrever um algoritmo que leia 10 valores para uma variável n e, para cada um deles, calcule a tabuada de 1 até n . Mostre a tabuada na forma: