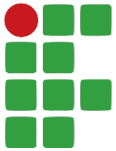


Algoritmos: Fluxo iterativo

PRG129001 – Programação I

Prof. Roberto Wanderley da Nóbrega
Instituto Federal de Santa Catarina

2024.1

Estes slides são baseados no material do Prof. Eraldo, disponível na wiki.

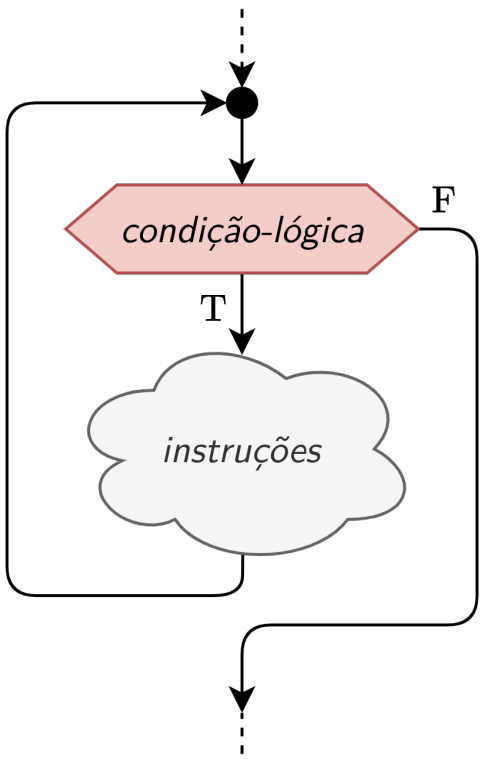
Conceito

Até então, vimos algoritmos em que cada comando é executado no máximo uma vez.

No entanto, uma das grandes vantagens de um sistema computacional é a capacidade de **repetir** um conjunto de instruções, sobre dados possivelmente diferentes, a uma velocidade muito grande.

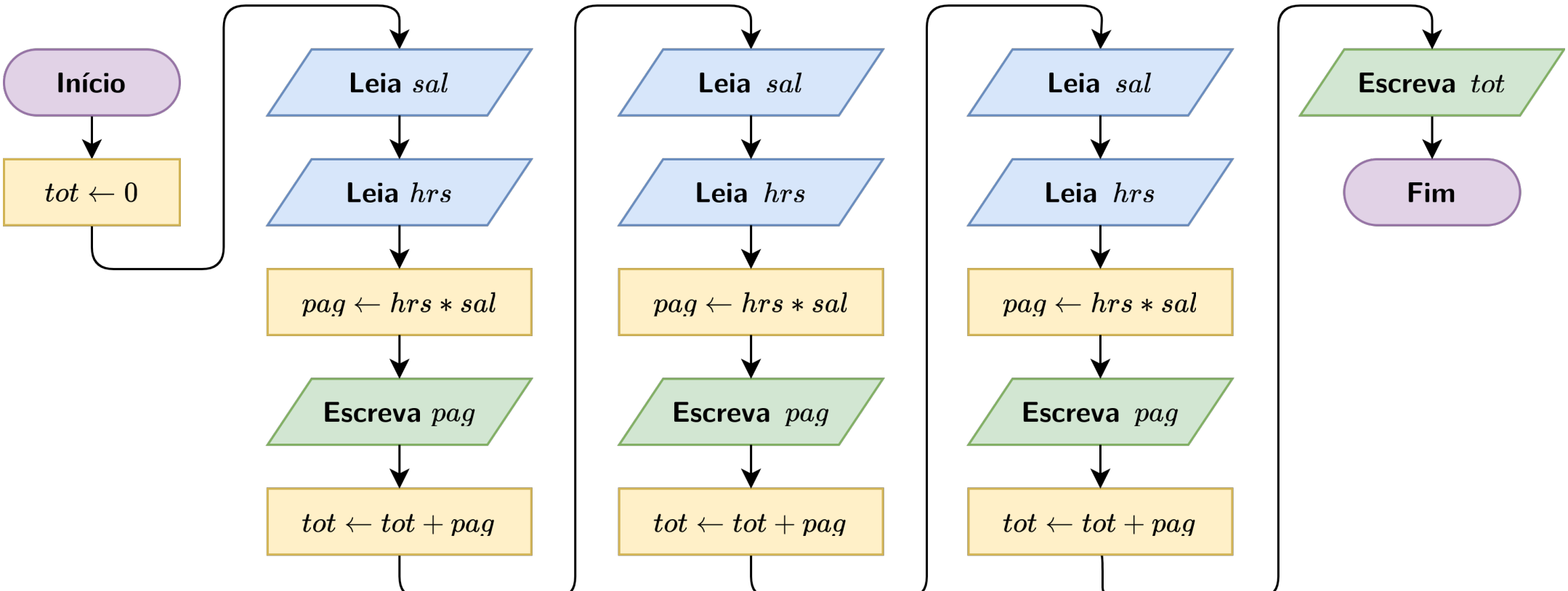
Comando Enquanto

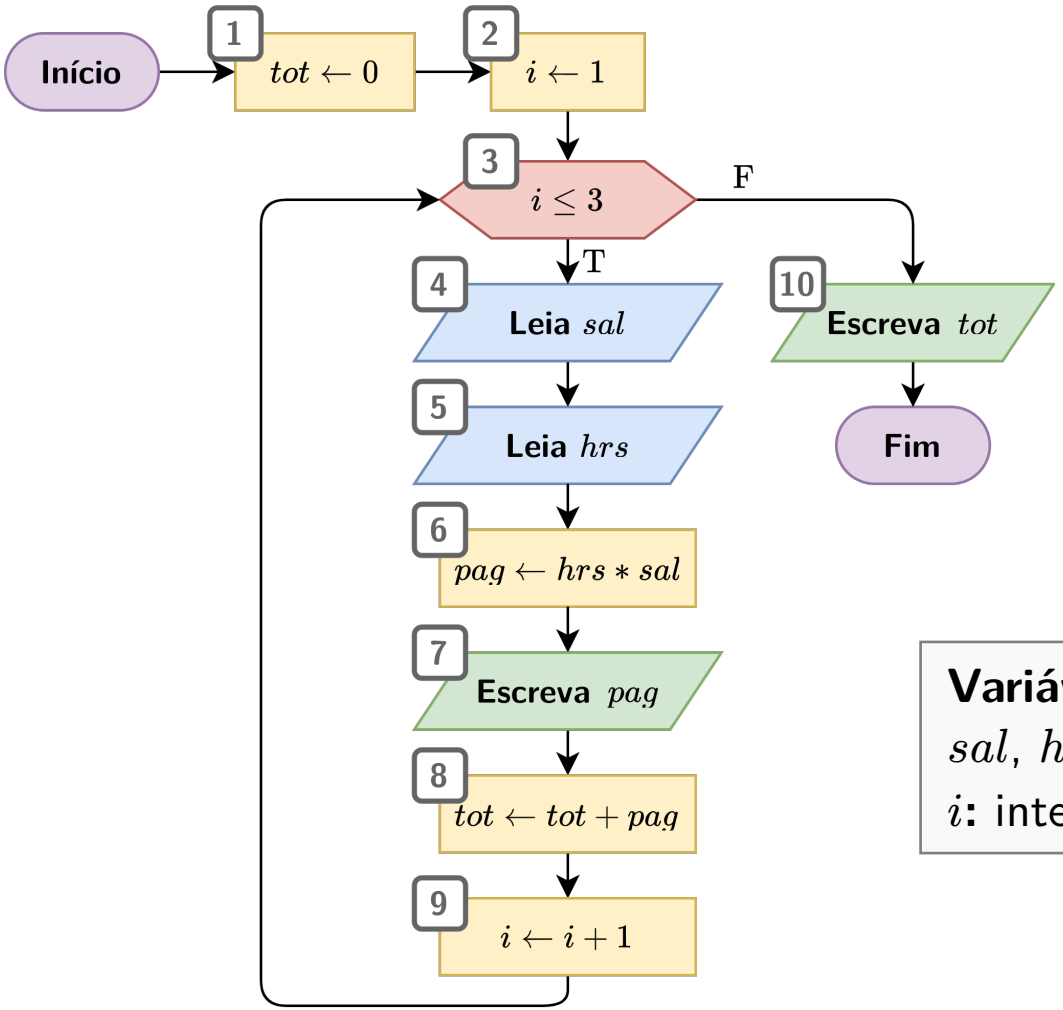
Enquanto *condição-lógica* **faça**
    instruções
FimEnquanto



Exemplo: Pagamentos

Elabore um algoritmo que lê os salários por hora (em R$ por hora) e as horas trabalhadas de três colaboradores de uma empresa e escreve o pagamento de cada um deles, bem como o total pago pela empresa.


Início
$tot \leftarrow 0$
Leia sal
Leia hrs
$pag \leftarrow hrs * sal$
Escreva pag
$tot \leftarrow tot + pag$
Leia sal
Leia hrs
$pag \leftarrow hrs * sal$
Escreva pag
$tot \leftarrow tot + pag$
Leia sal
Leia hrs
$pag \leftarrow hrs * sal$
Escreva pag
$tot \leftarrow tot + pag$
Escreva tot
Fim

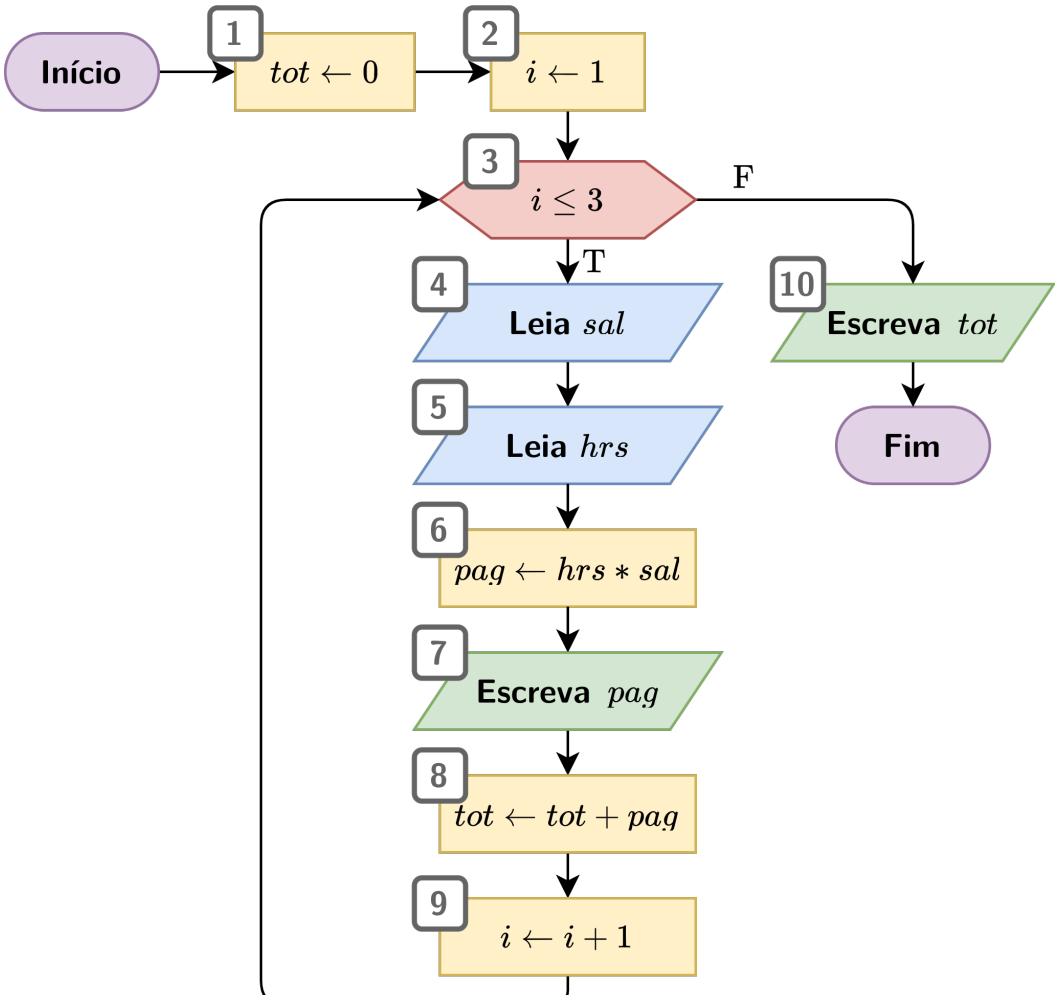


Variáveis

sal, hrs, pag, tot: real

i: inteiro

```
1  Início
2      tot ← 0
3      i ← 1
4      Enquanto i ≤ 3 faça
5          Leia sal
6          Leia hrs
7          pag ← hrs * sal
8          Escreva pag
9          tot ← tot + pag
10         i ← i + 1
11     FimEnquanto
12     Escreva tot
13 Fim
```

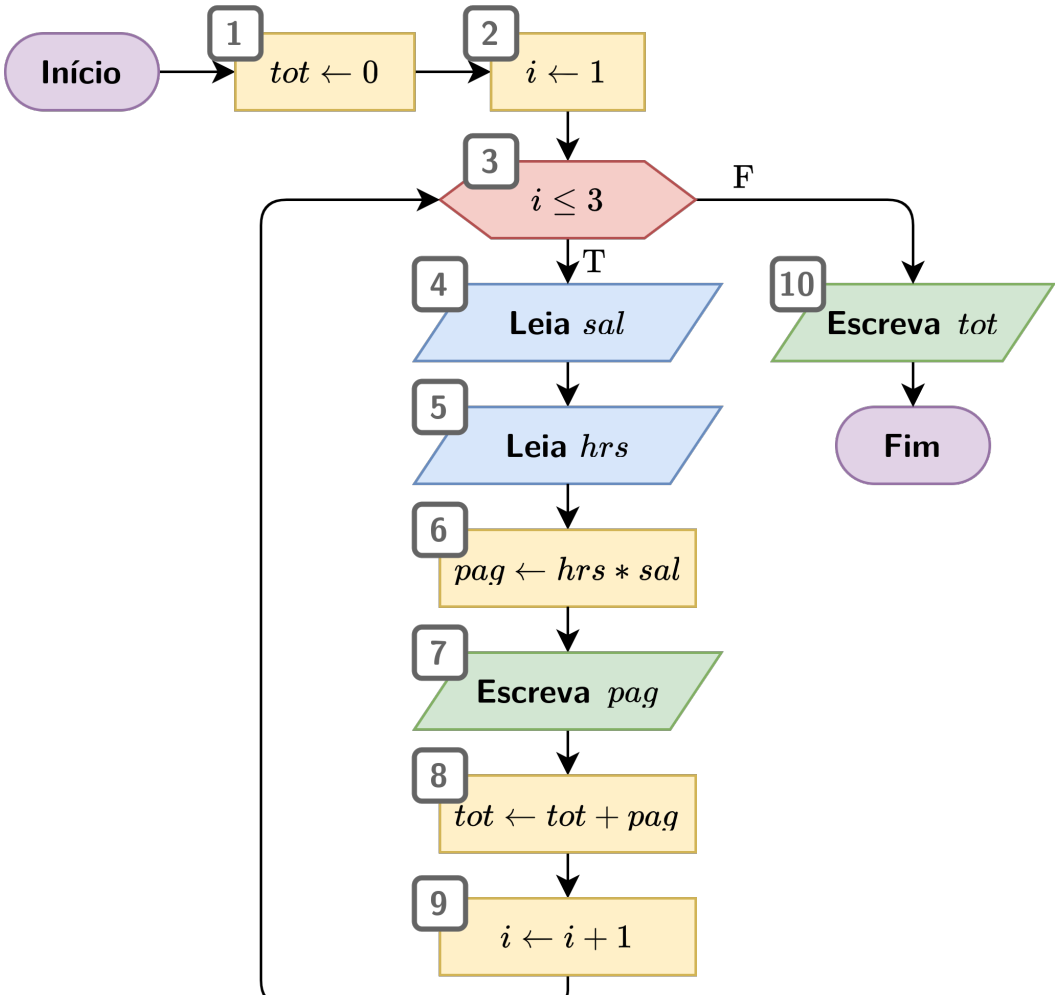


Entrada: `50 8 60 7 50 7.5`

Saída:

Instr	sal	hrs	pag	i	tot
Início	?	?	?	?	?

Instr	sal	hrs	pag	i	tot

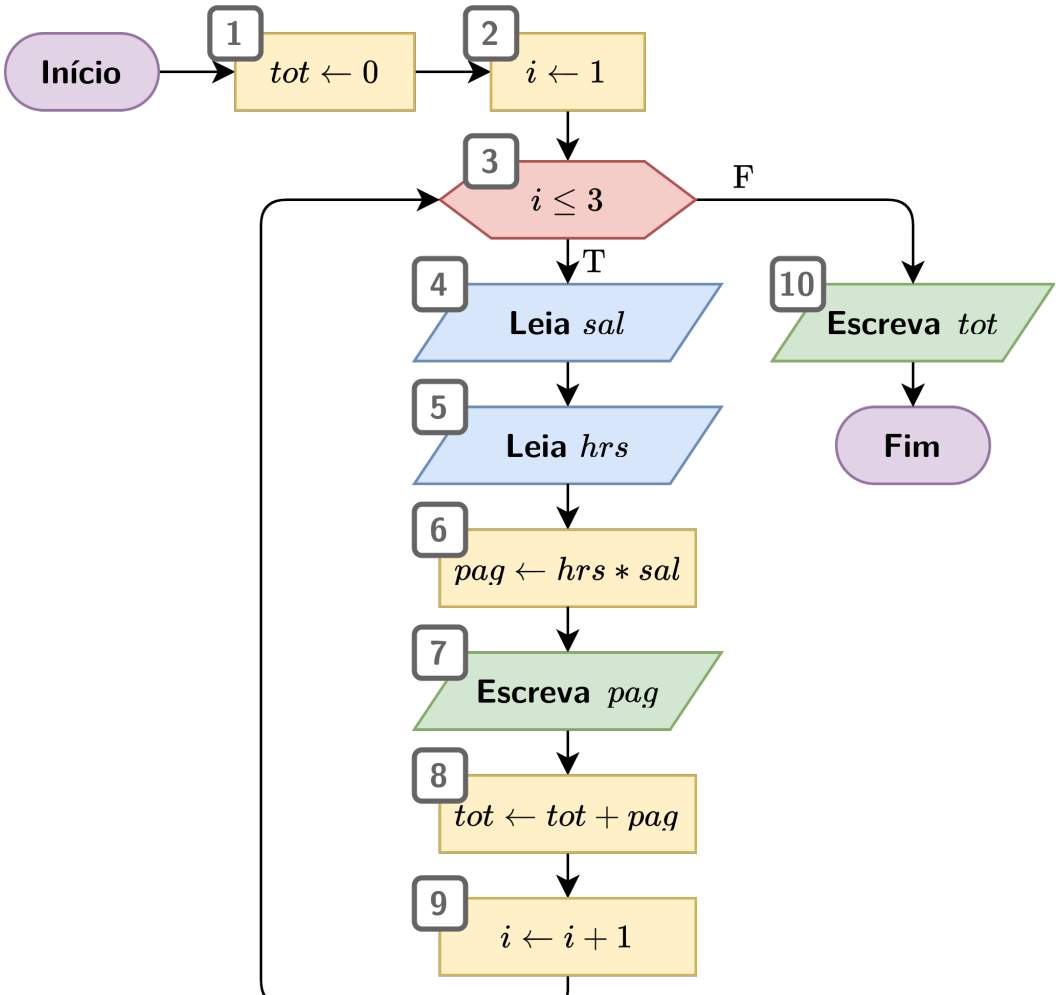


Entrada: `50 8 60 7 50 7.5`

Saída:

Instr	sal	hrs	pag	i	tot
Início	?	?	?	?	?
1	?	?	?	?	0

Instr	sal	hrs	pag	i	tot

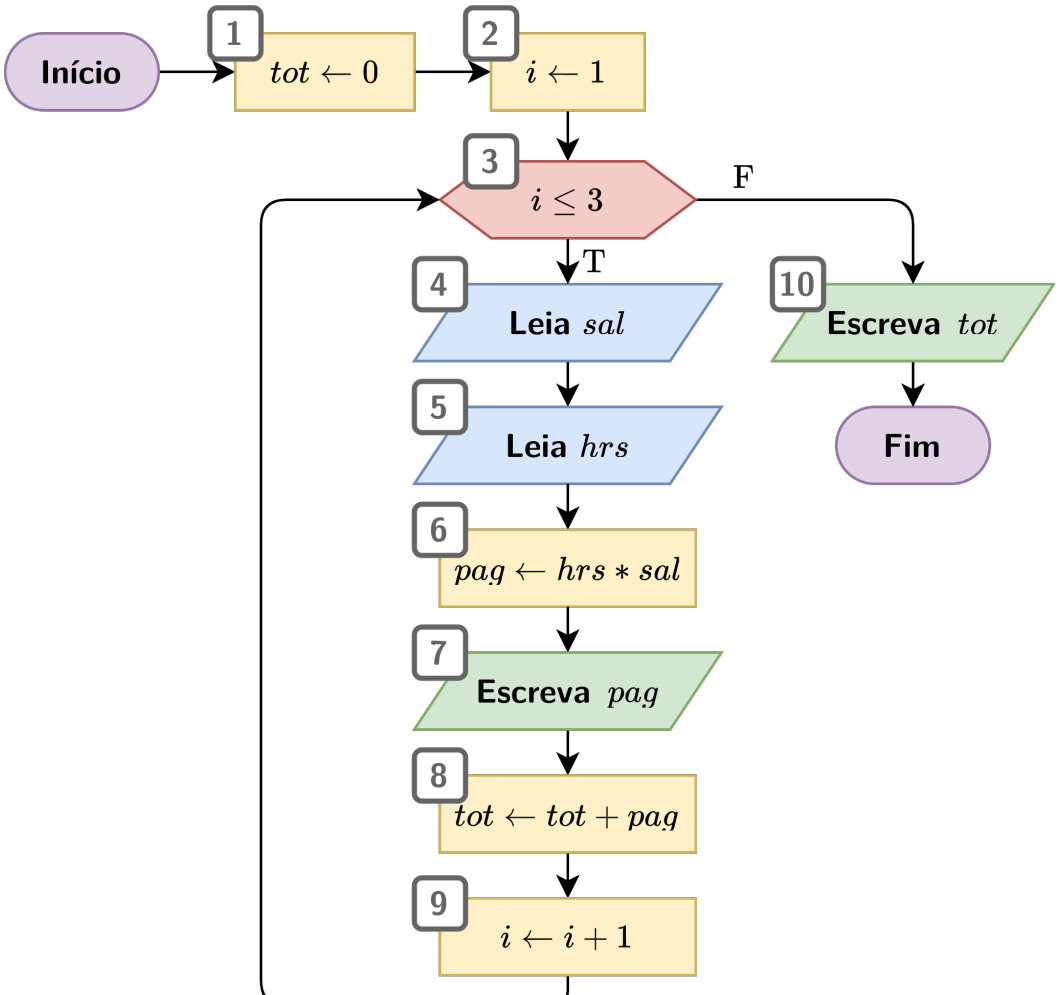


Entrada: `50 8 60 7 50 7.5`

Saída:

Instr	sal	hrs	pag	i	tot
Início	?	?	?	?	?
1	?	?	?	?	0
2	?	?	?	1	0

Instr	sal	hrs	pag	i	tot

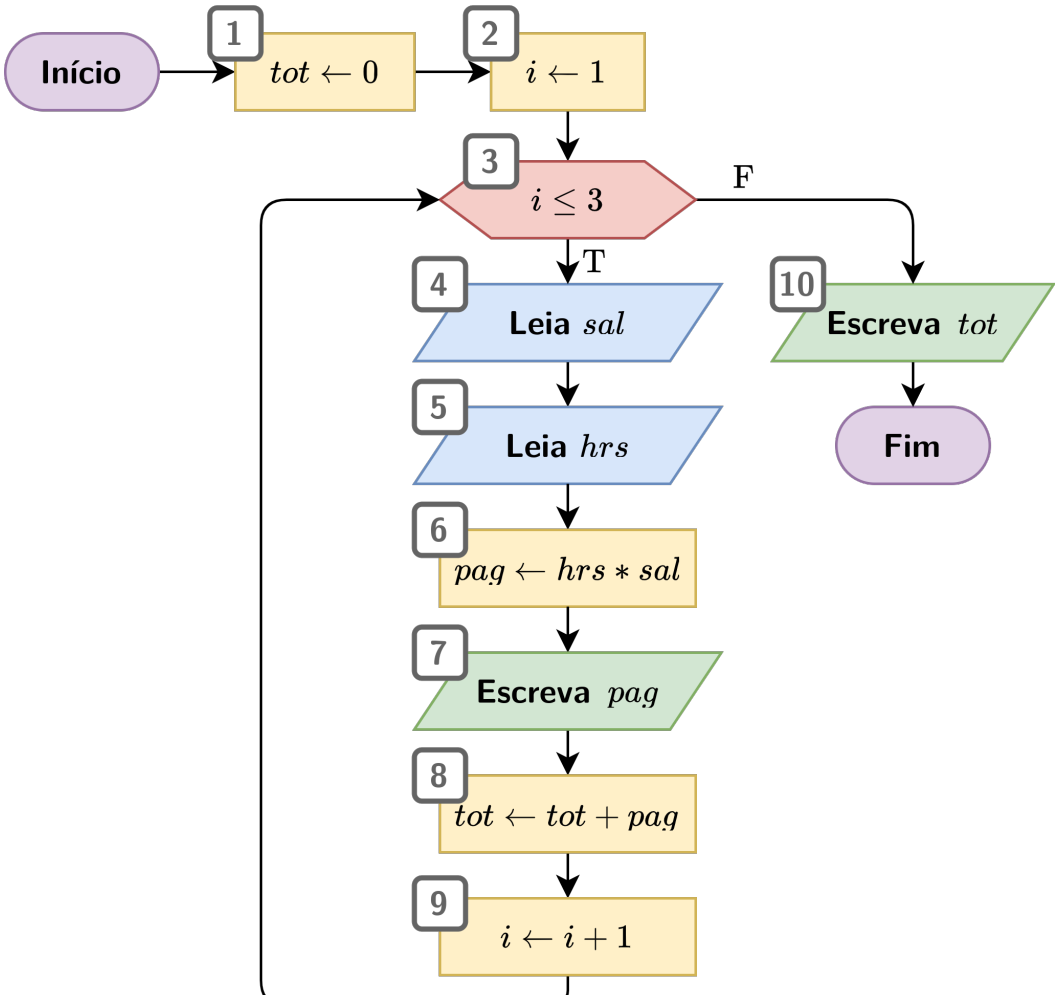


Entrada: `50 8 60 7 50 7.5`

Saída:

Instr	sal	hrs	pag	i	tot
Início	?	?	?	?	?
1	?	?	?	?	0
2	?	?	?	1	0
3	?	?	?	1	0

Instr	sal	hrs	pag	i	tot

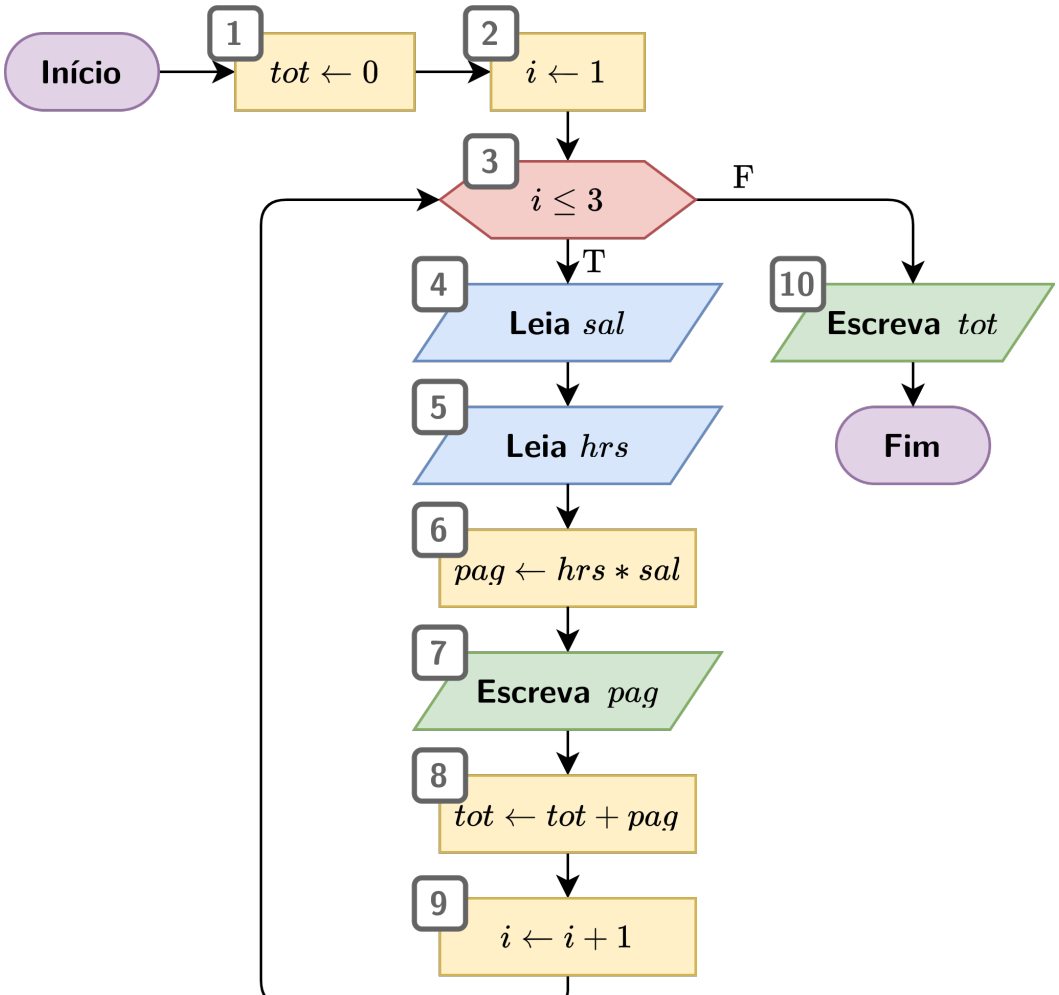


Entrada: ~~50~~ 8 60 7 50 7.5

Saída:

Instr	sal	hrs	pag	i	tot
Início	?	?	?	?	?
1	?	?	?	?	0
2	?	?	?	1	0
3	?	?	?	1	0
4	50	?	?	1	0

Instr	sal	hrs	pag	i	tot

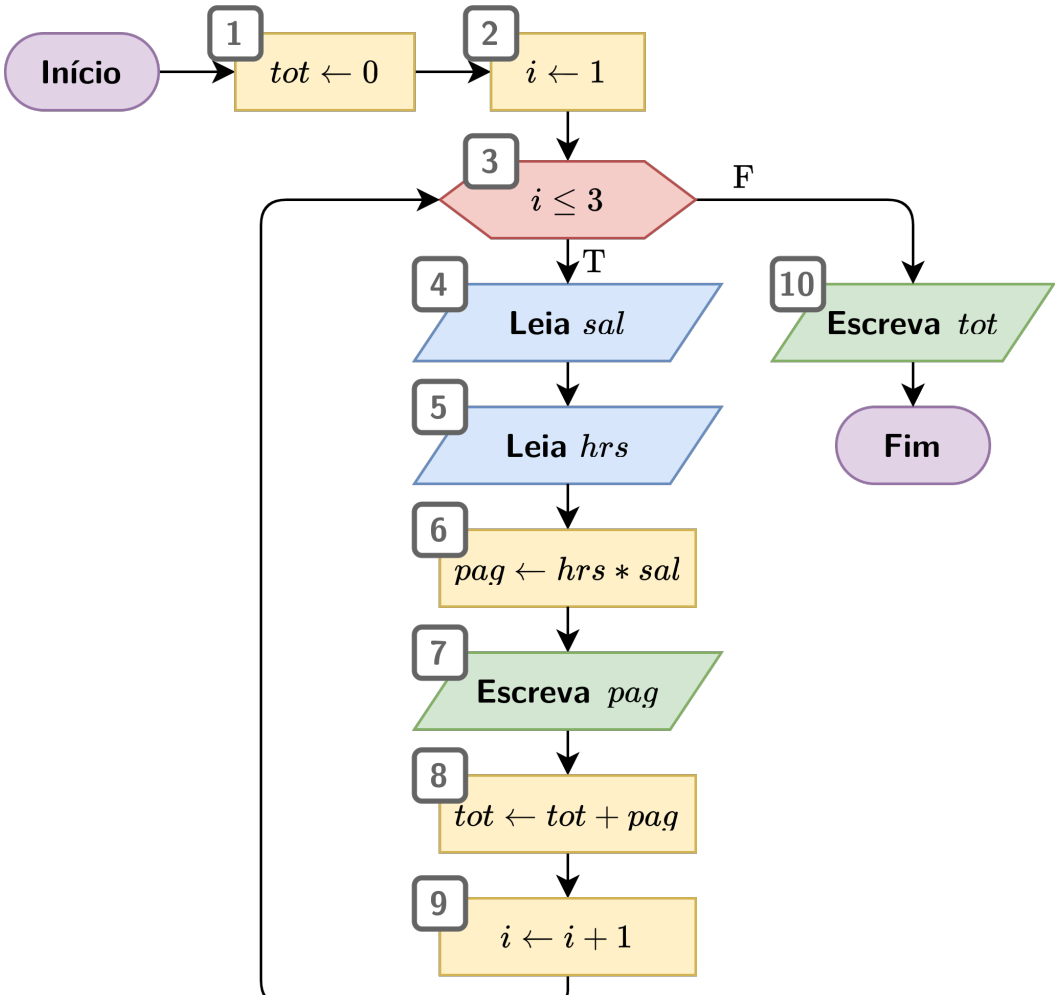


Entrada: ~~50~~ ~~8~~ 60 7 50 7.5

Saída:

Instr	sal	hrs	pag	i	tot
Início	?	?	?	?	?
1	?	?	?	?	0
2	?	?	?	1	0
3	?	?	?	1	0
4	50	?	?	1	0
5	50	8	?	1	0

Instr	sal	hrs	pag	i	tot

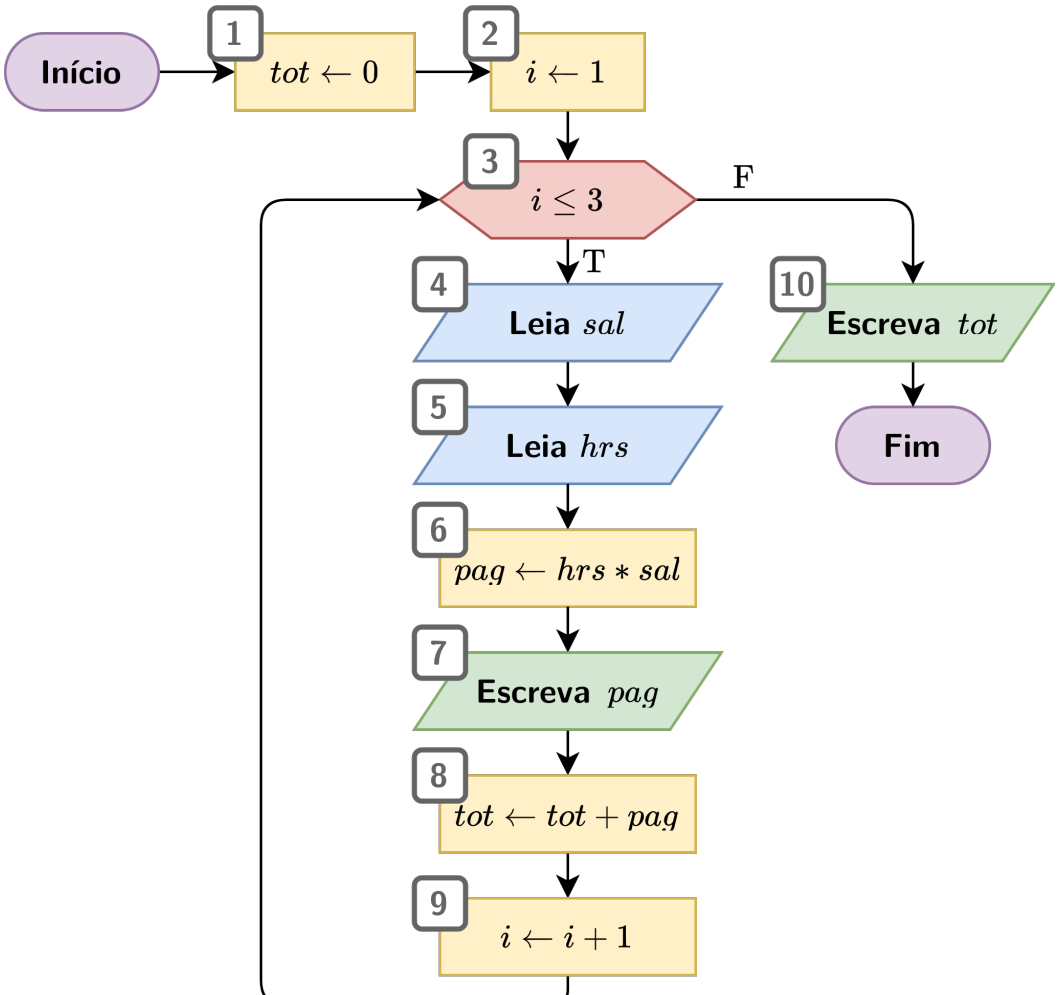


Entrada: ~~50~~ ~~8~~ 60 7 50 7.5

Saída:

Instr	sal	hrs	pag	i	tot
Início	?	?	?	?	?
1	?	?	?	?	0
2	?	?	?	1	0
3	?	?	?	1	0
4	50	?	?	1	0
5	50	8	?	1	0
6	50	8	400	1	0

Instr	sal	hrs	pag	i	tot

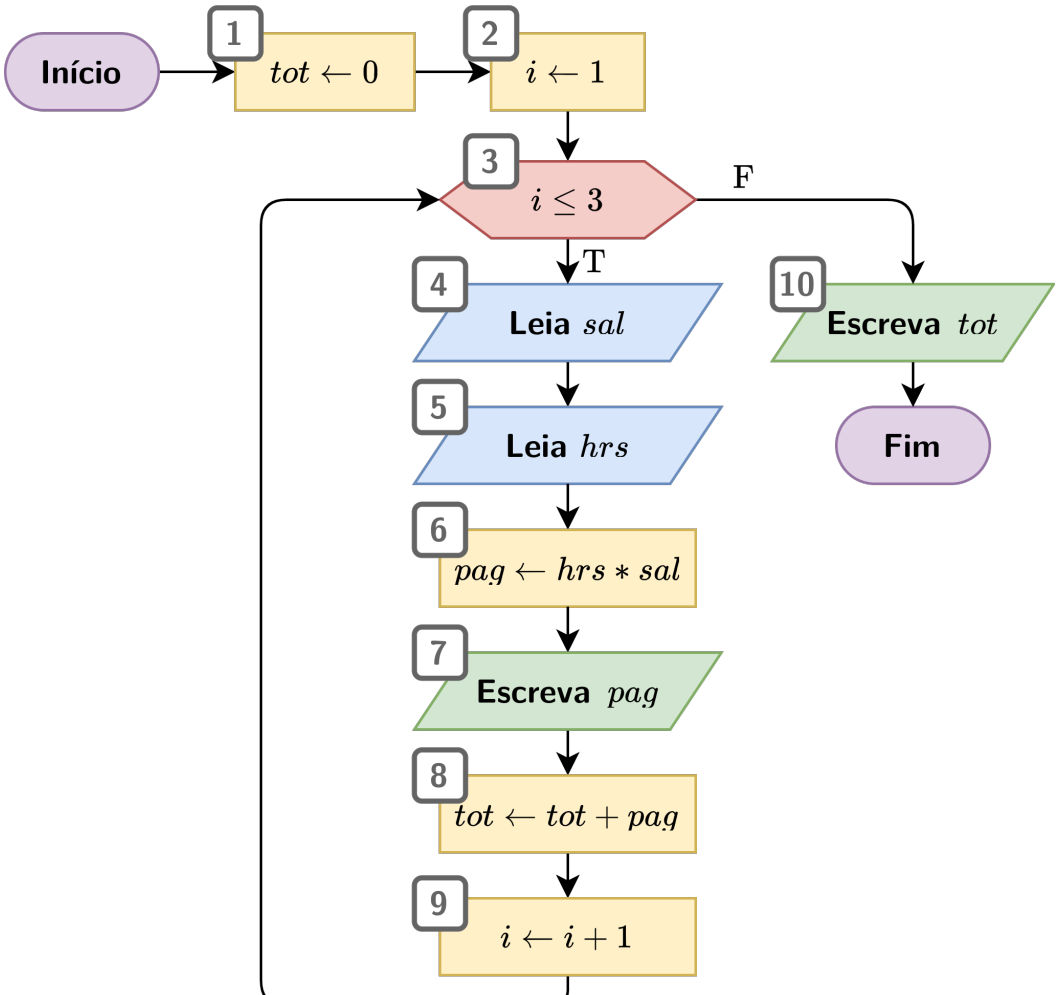


Entrada: ~~50~~ ~~8~~ 60 7 50 7.5

Saída: 400

Instr	sal	hrs	pag	i	tot
Início	?	?	?	?	?
1	?	?	?	?	0
2	?	?	?	1	0
3	?	?	?	1	0
4	50	?	?	1	0
5	50	8	?	1	0
6	50	8	400	1	0
7	50	8	400	1	0

Instr	sal	hrs	pag	i	tot

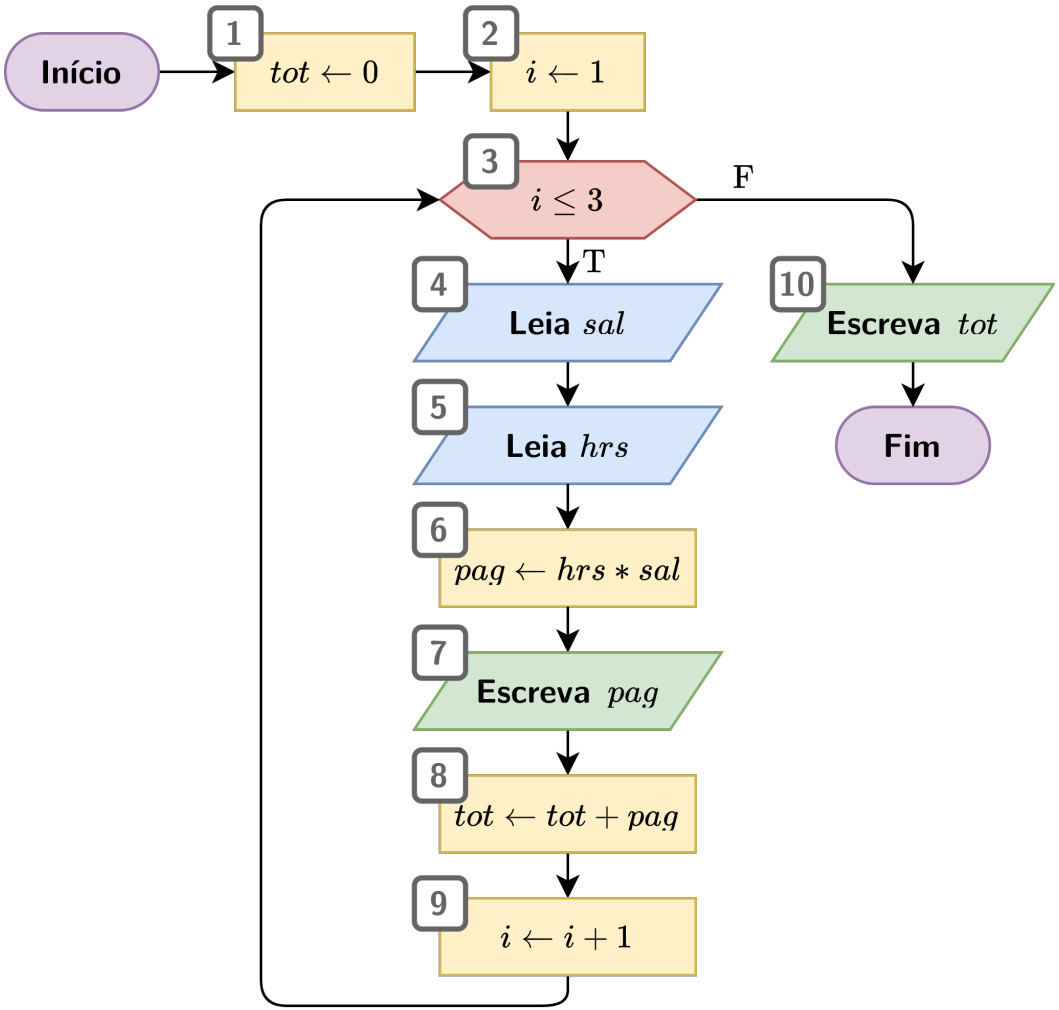


Entrada: ~~50~~ ~~8~~ 60 7 50 7.5

Saída: 400

Instr	sal	hrs	pag	i	tot
Início	?	?	?	?	?
1	?	?	?	?	0
2	?	?	?	1	0
3	?	?	?	1	0
4	50	?	?	1	0
5	50	8	?	1	0
6	50	8	400	1	0
7	50	8	400	1	0
8	50	8	400	1	400

Instr	sal	hrs	pag	i	tot

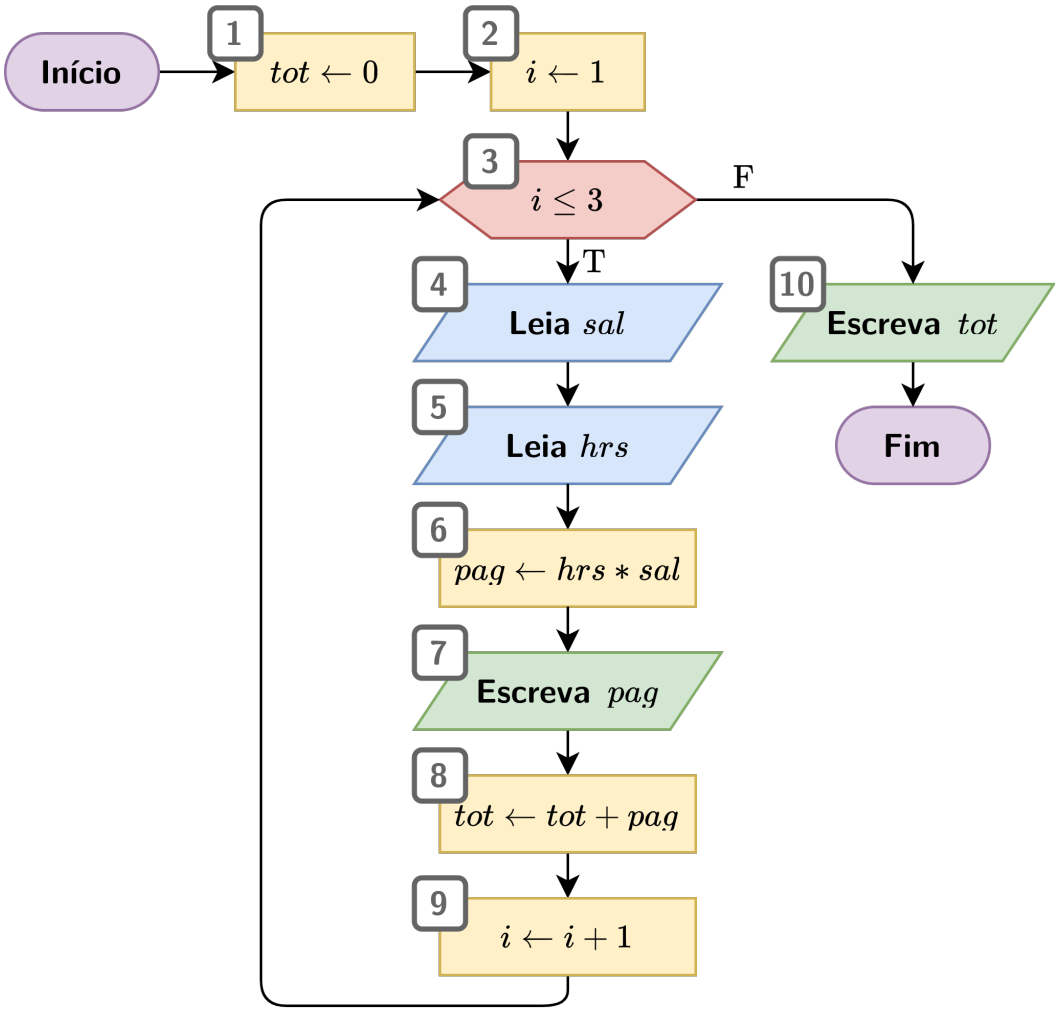


Entrada: ~~50~~ ~~8~~ 60 7 50 7.5

Saída: 400

Instr	sal	hrs	pag	i	tot
Início	?	?	?	?	?
1	?	?	?	?	0
2	?	?	?	1	0
3	?	?	?	1	0
4	50	?	?	1	0
5	50	8	?	1	0
6	50	8	400	1	0
7	50	8	400	1	0
8	50	8	400	1	400
9	50	8	400	2	400

Instr	sal	hrs	pag	i	tot

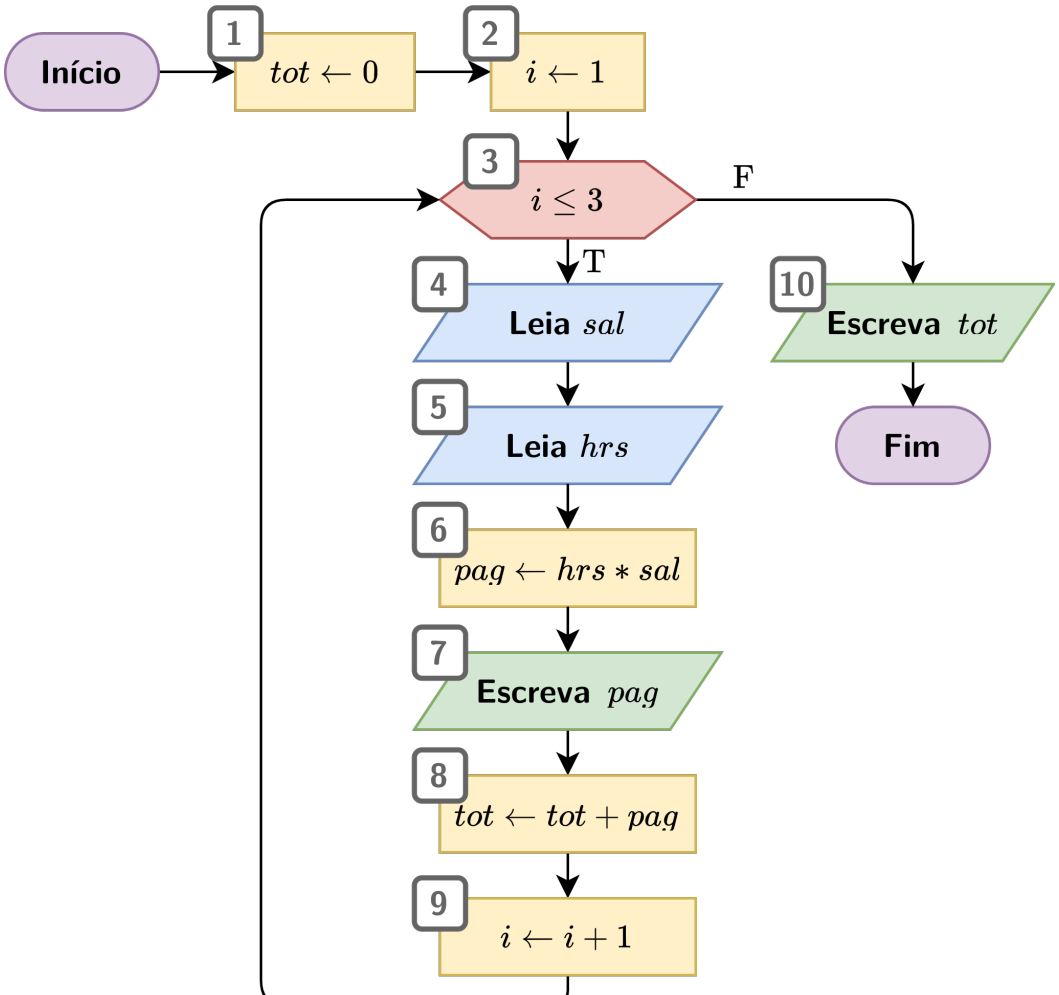


Entrada: ~~50~~ ~~8~~ 60 7 50 7.5

Saída: 400

Instr	sal	hrs	pag	i	tot
Início	?	?	?	?	?
1	?	?	?	?	0
2	?	?	?	1	0
3	?	?	?	1	0
4	50	?	?	1	0
5	50	8	?	1	0
6	50	8	400	1	0
7	50	8	400	1	0
8	50	8	400	1	400
9	50	8	400	2	400
3	50	8	400	2	400

Instr	sal	hrs	pag	i	tot

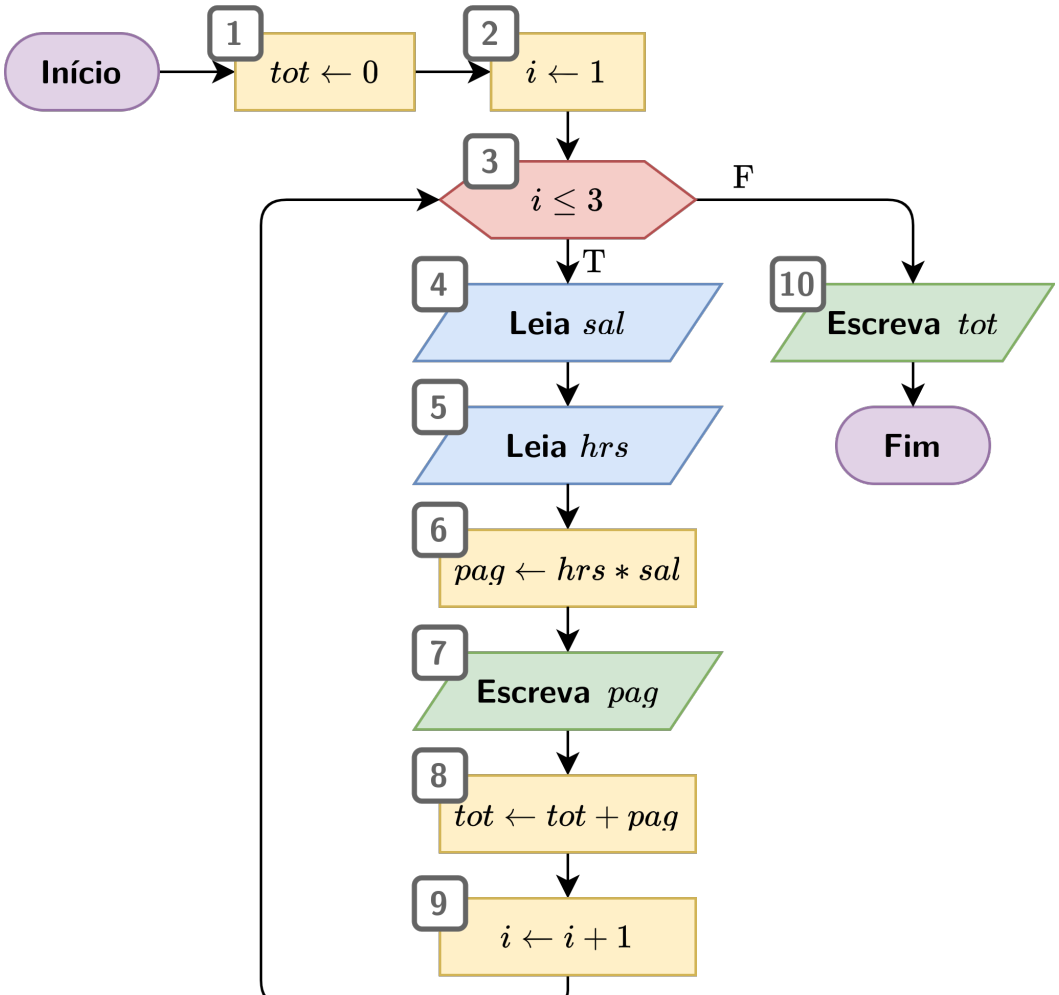


Entrada: ~~50~~ ~~8~~ ~~60~~ 7 50 7.5

Saída: 400

Instr	sal	hrs	pag	i	tot
Início	?	?	?	?	?
1	?	?	?	?	0
2	?	?	?	1	0
3	?	?	?	1	0
4	50	?	?	1	0
5	50	8	?	1	0
6	50	8	400	1	0
7	50	8	400	1	0
8	50	8	400	1	400
9	50	8	400	2	400
3	50	8	400	2	400
4	60	8	400	2	400

Instr	sal	hrs	pag	i	tot

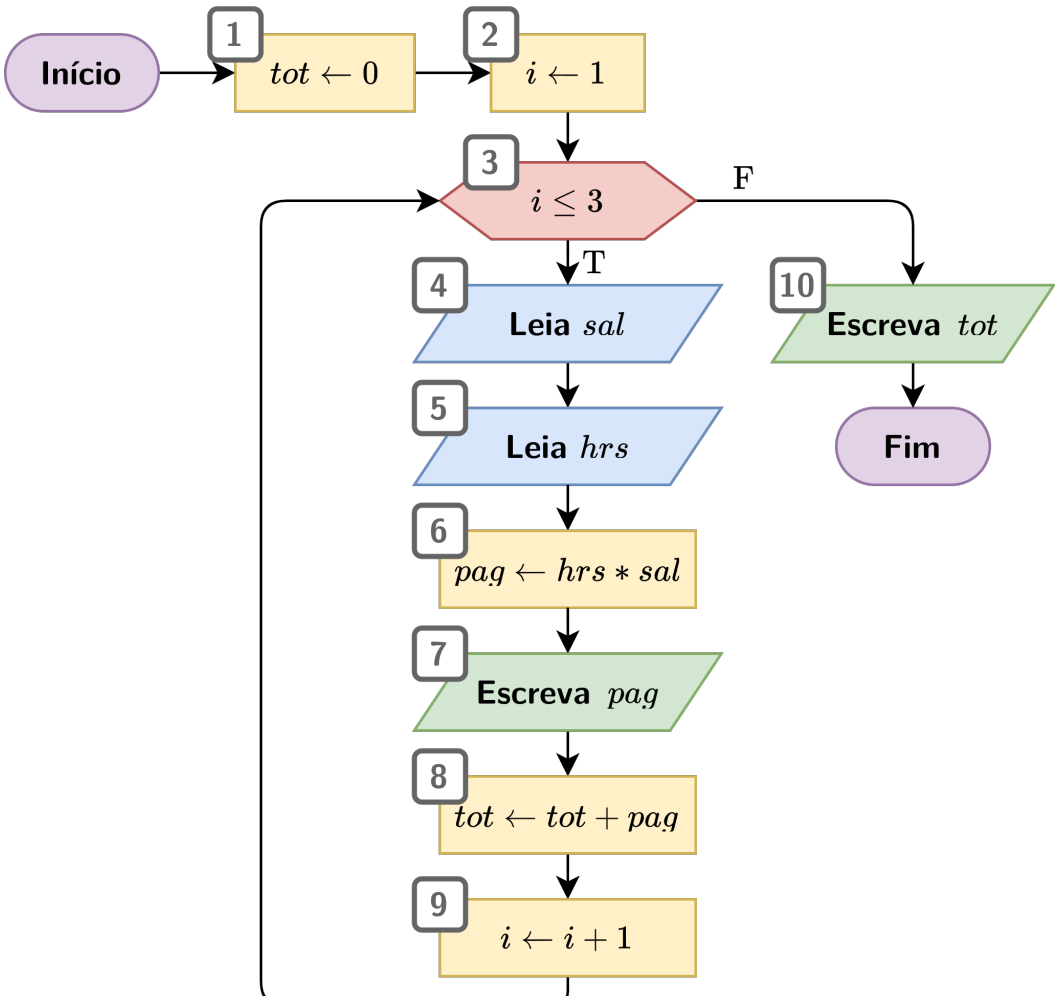


Entrada: ~~50~~ ~~8~~ ~~60~~ ~~7~~ 50 7.5

Saída: 400

Instr	sal	hrs	pag	i	tot
Início	?	?	?	?	?
1	?	?	?	?	0
2	?	?	?	1	0
3	?	?	?	1	0
4	50	?	?	1	0
5	50	8	?	1	0
6	50	8	400	1	0
7	50	8	400	1	0
8	50	8	400	1	400
9	50	8	400	2	400
3	50	8	400	2	400
4	60	8	400	2	400
5	60	7	400	2	400

Instr	sal	hrs	pag	i	tot

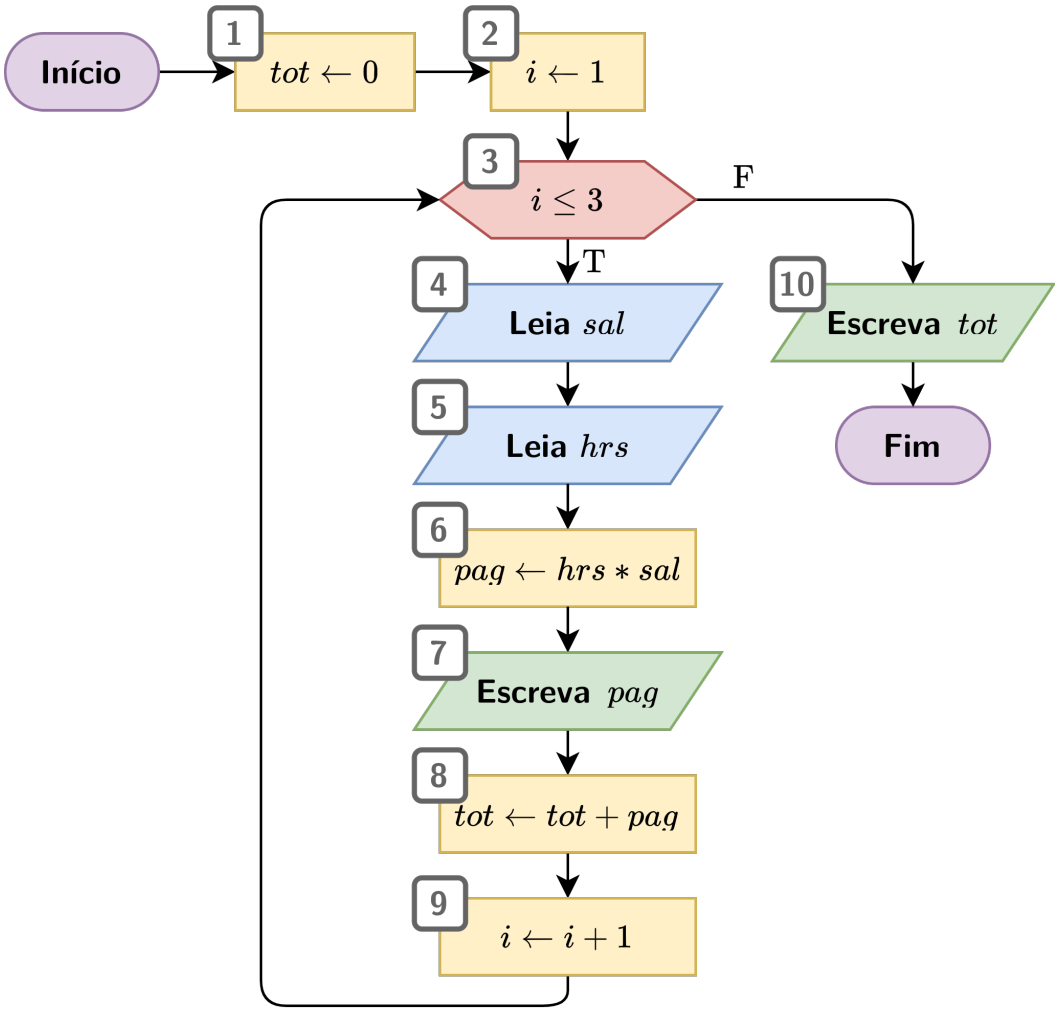


Entrada: ~~50~~ ~~8~~ ~~60~~ ~~7~~ 50 7.5

Saída: 400

Instr	sal	hrs	pag	i	tot
Início	?	?	?	?	?
1	?	?	?	?	0
2	?	?	?	1	0
3	?	?	?	1	0
4	50	?	?	1	0
5	50	8	?	1	0
6	50	8	400	1	0
7	50	8	400	1	0
8	50	8	400	1	400
9	50	8	400	2	400
3	50	8	400	2	400
4	60	8	400	2	400
5	60	7	400	2	400

Instr	sal	hrs	pag	i	tot
6	60	7	420	2	400

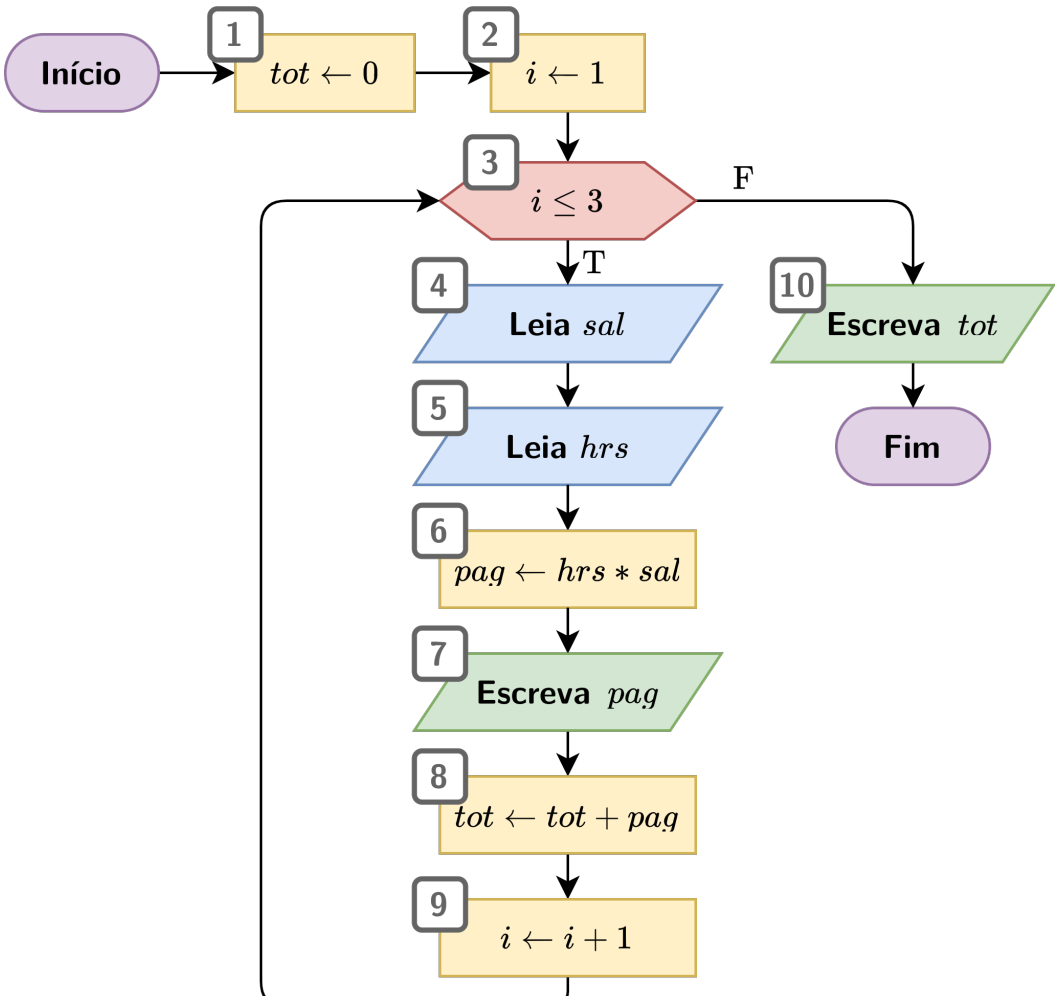


Entrada: ~~50~~ ~~8~~ ~~60~~ ~~7~~ 50 7.5

Saída: 400 420

Instr	sal	hrs	pag	i	tot
Início	?	?	?	?	?
1	?	?	?	?	0
2	?	?	?	1	0
3	?	?	?	1	0
4	50	?	?	1	0
5	50	8	?	1	0
6	50	8	400	1	0
7	50	8	400	1	0
8	50	8	400	1	400
9	50	8	400	2	400
3	50	8	400	2	400
4	60	8	400	2	400
5	60	7	400	2	400

Instr	sal	hrs	pag	i	tot
6	60	7	420	2	400
7	60	7	420	2	400

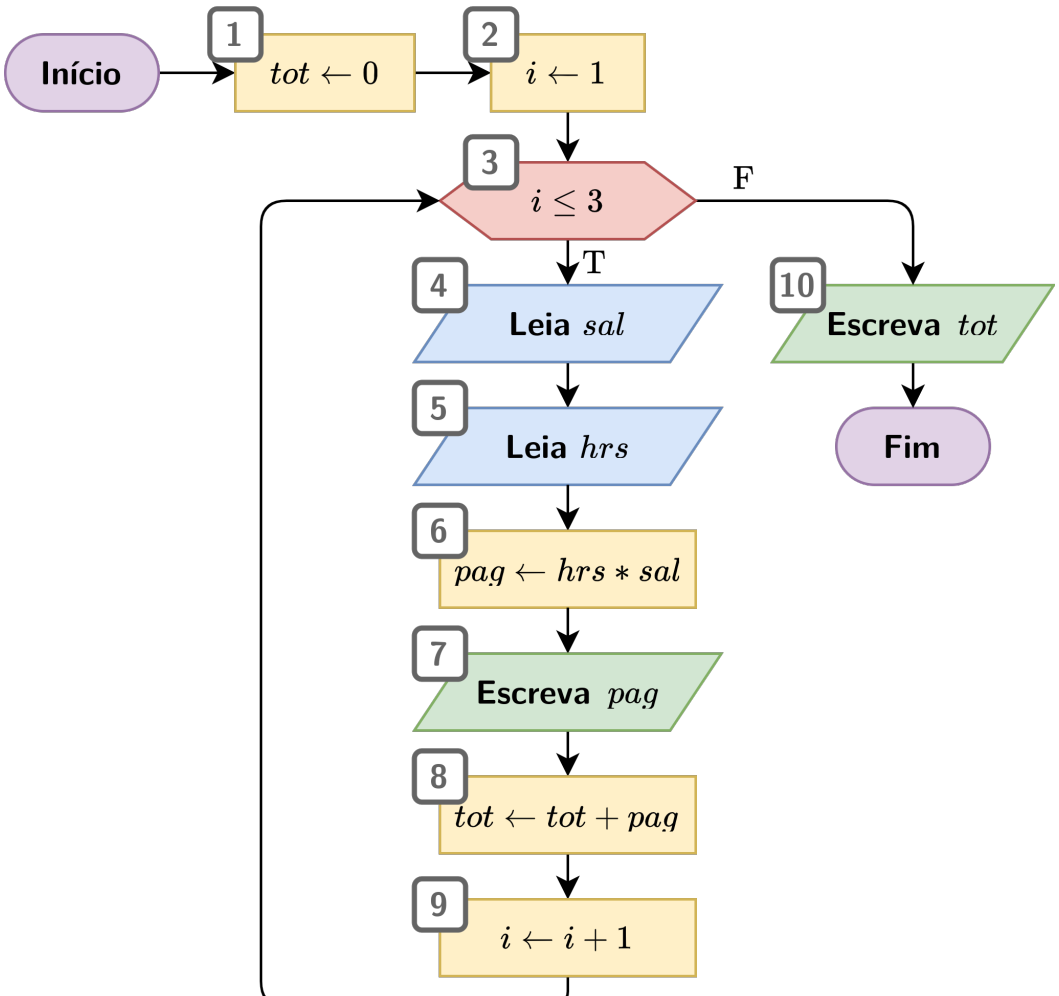


Entrada: ~~50~~ ~~8~~ ~~60~~ ~~7~~ 50 7.5

Saída: 400 420

Instr	sal	hrs	pag	i	tot
Início	?	?	?	?	?
1	?	?	?	?	0
2	?	?	?	1	0
3	?	?	?	1	0
4	50	?	?	1	0
5	50	8	?	1	0
6	50	8	400	1	0
7	50	8	400	1	0
8	50	8	400	1	400
9	50	8	400	2	400
3	50	8	400	2	400
4	60	8	400	2	400
5	60	7	400	2	400

Instr	sal	hrs	pag	i	tot
6	60	7	420	2	400
7	60	7	420	2	400
8	60	7	420	2	820

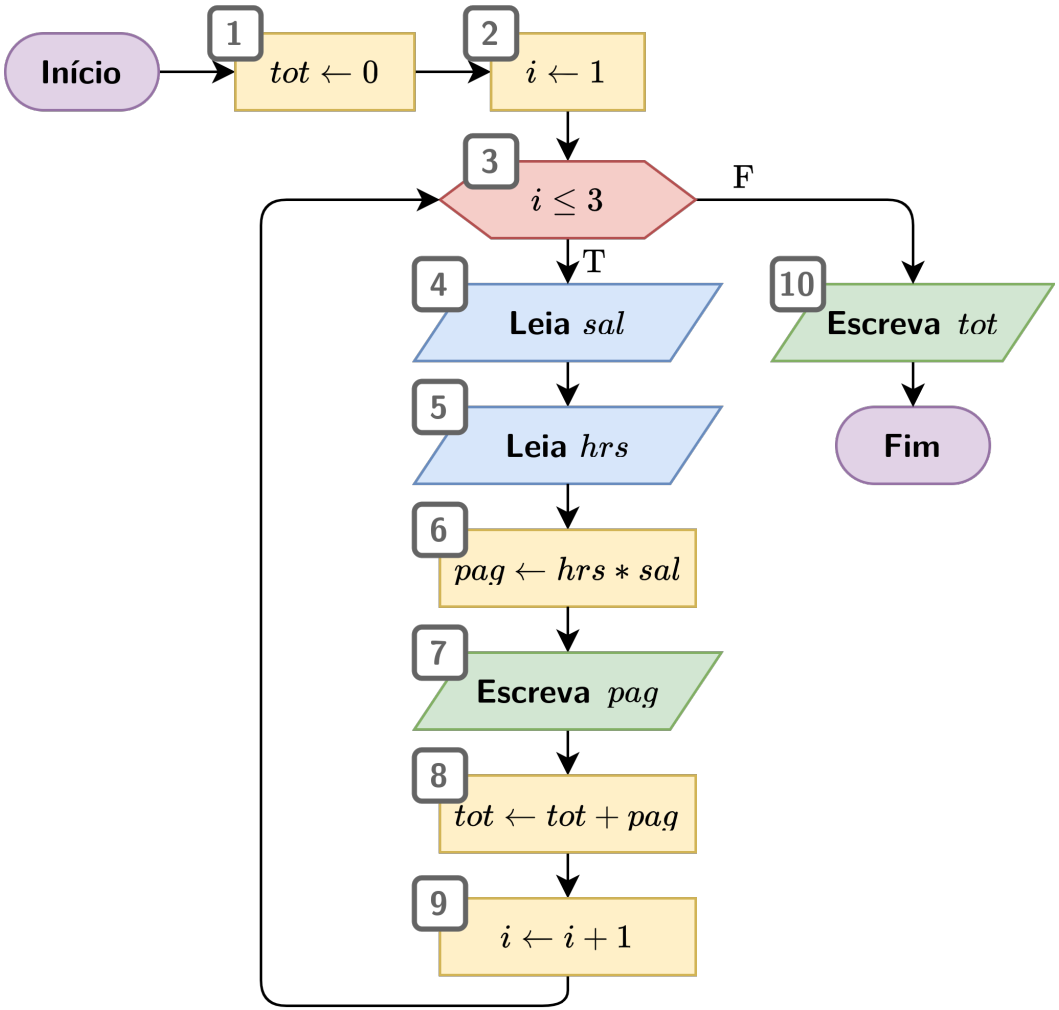


Entrada: ~~50~~ ~~8~~ ~~60~~ ~~7~~ 50 7.5

Saída: 400 420

Instr	sal	hrs	pag	i	tot
Início	?	?	?	?	?
1	?	?	?	?	0
2	?	?	?	1	0
3	?	?	?	1	0
4	50	?	?	1	0
5	50	8	?	1	0
6	50	8	400	1	0
7	50	8	400	1	0
8	50	8	400	1	400
9	50	8	400	2	400
3	50	8	400	2	400
4	60	8	400	2	400
5	60	7	400	2	400

Instr	sal	hrs	pag	i	tot
6	60	7	420	2	400
7	60	7	420	2	400
8	60	7	420	2	820
9	60	7	420	3	820

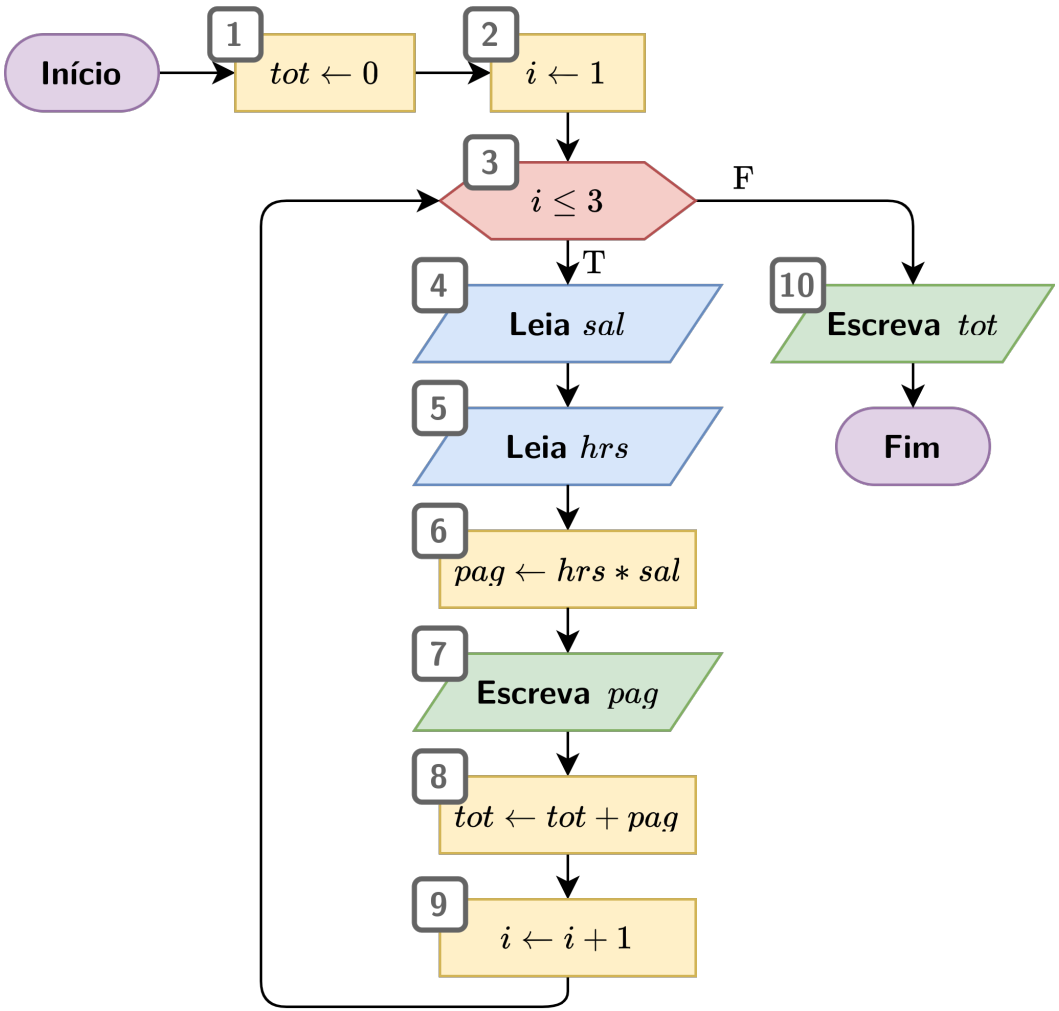


Entrada: ~~50~~ ~~8~~ ~~60~~ ~~7~~ 50 7.5

Saída: 400 420

Instr	sal	hrs	pag	i	tot
Início	?	?	?	?	?
1	?	?	?	?	0
2	?	?	?	1	0
3	?	?	?	1	0
4	50	?	?	1	0
5	50	8	?	1	0
6	50	8	400	1	0
7	50	8	400	1	0
8	50	8	400	1	400
9	50	8	400	2	400
3	50	8	400	2	400
4	60	8	400	2	400
5	60	7	400	2	400

Instr	sal	hrs	pag	i	tot
6	60	7	420	2	400
7	60	7	420	2	400
8	60	7	420	2	820
9	60	7	420	3	820
3	60	7	420	3	820

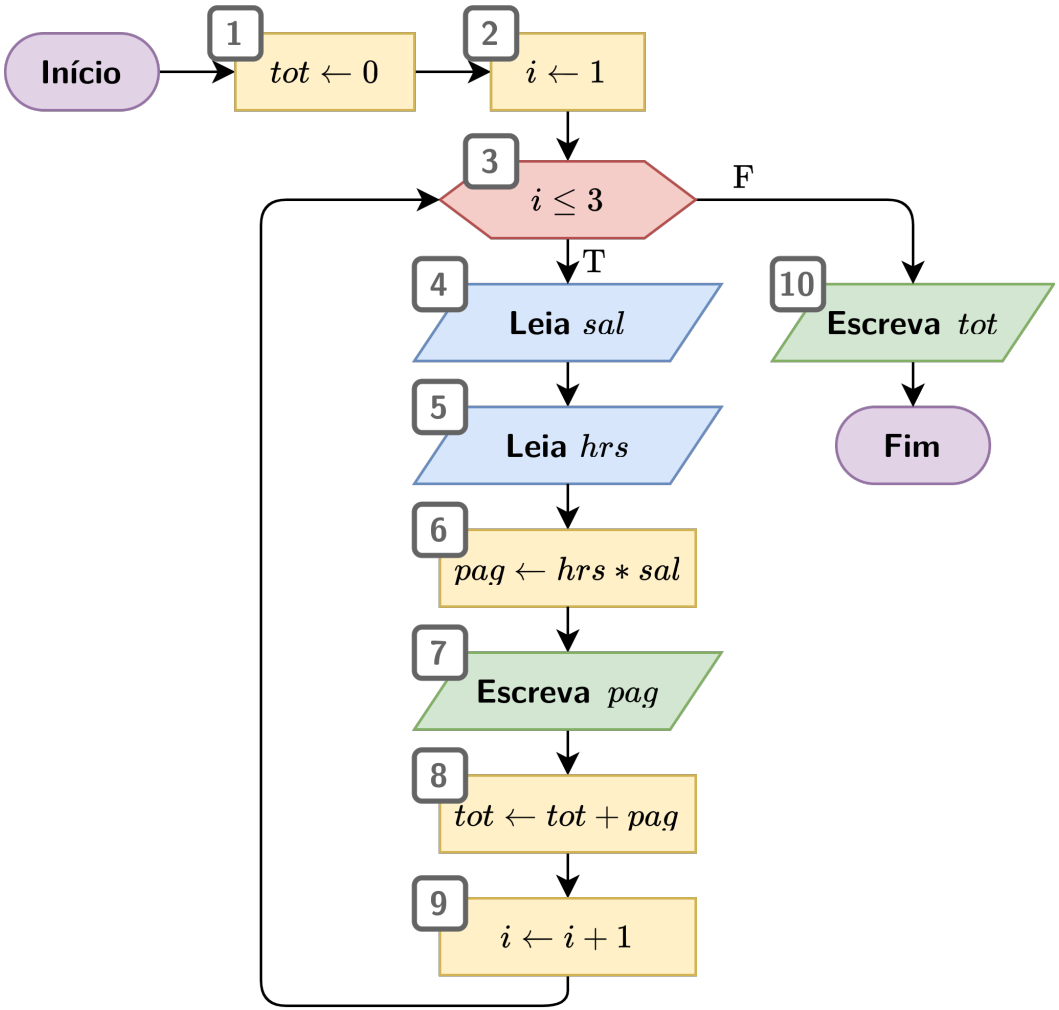


Entrada: ~~50~~ ~~8~~ ~~60~~ ~~7~~ ~~50~~ 7.5

Saída: 400 420

Instr	sal	hrs	pag	i	tot
Início	?	?	?	?	?
1	?	?	?	?	0
2	?	?	?	1	0
3	?	?	?	1	0
4	50	?	?	1	0
5	50	8	?	1	0
6	50	8	400	1	0
7	50	8	400	1	0
8	50	8	400	1	400
9	50	8	400	2	400
3	50	8	400	2	400
4	60	8	400	2	400
5	60	7	400	2	400

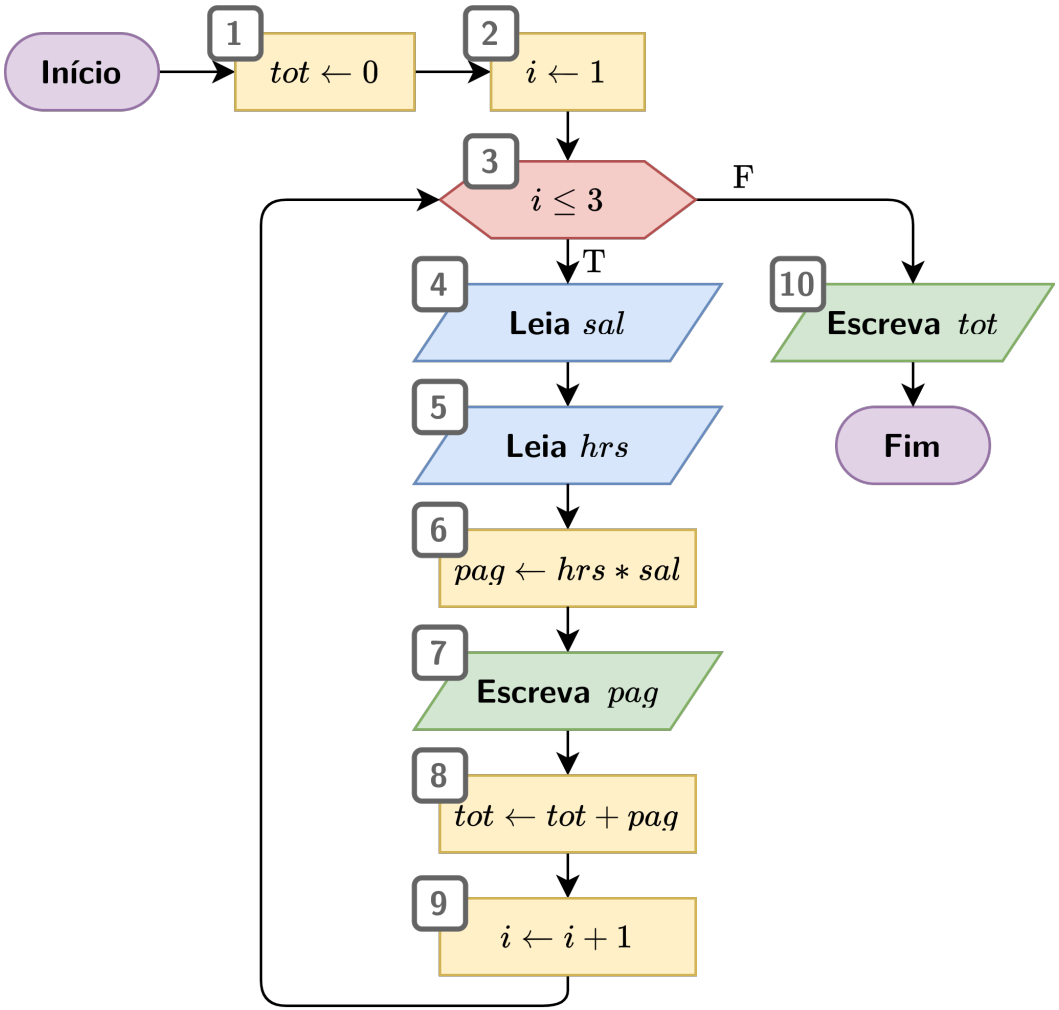
Instr	sal	hrs	pag	i	tot
6	60	7	420	2	400
7	60	7	420	2	400
8	60	7	420	2	820
9	60	7	420	3	820
3	60	7	420	3	820
4	50	7	420	3	820



Entrada: ~~50 8 60 7 50 7.5~~ Saída: 400 420

Instr	sal	hrs	pag	i	tot
Início	?	?	?	?	?
1	?	?	?	?	0
2	?	?	?	1	0
3	?	?	?	1	0
4	50	?	?	1	0
5	50	8	?	1	0
6	50	8	400	1	0
7	50	8	400	1	0
8	50	8	400	1	400
9	50	8	400	2	400
3	50	8	400	2	400
4	60	8	400	2	400
5	60	7	400	2	400

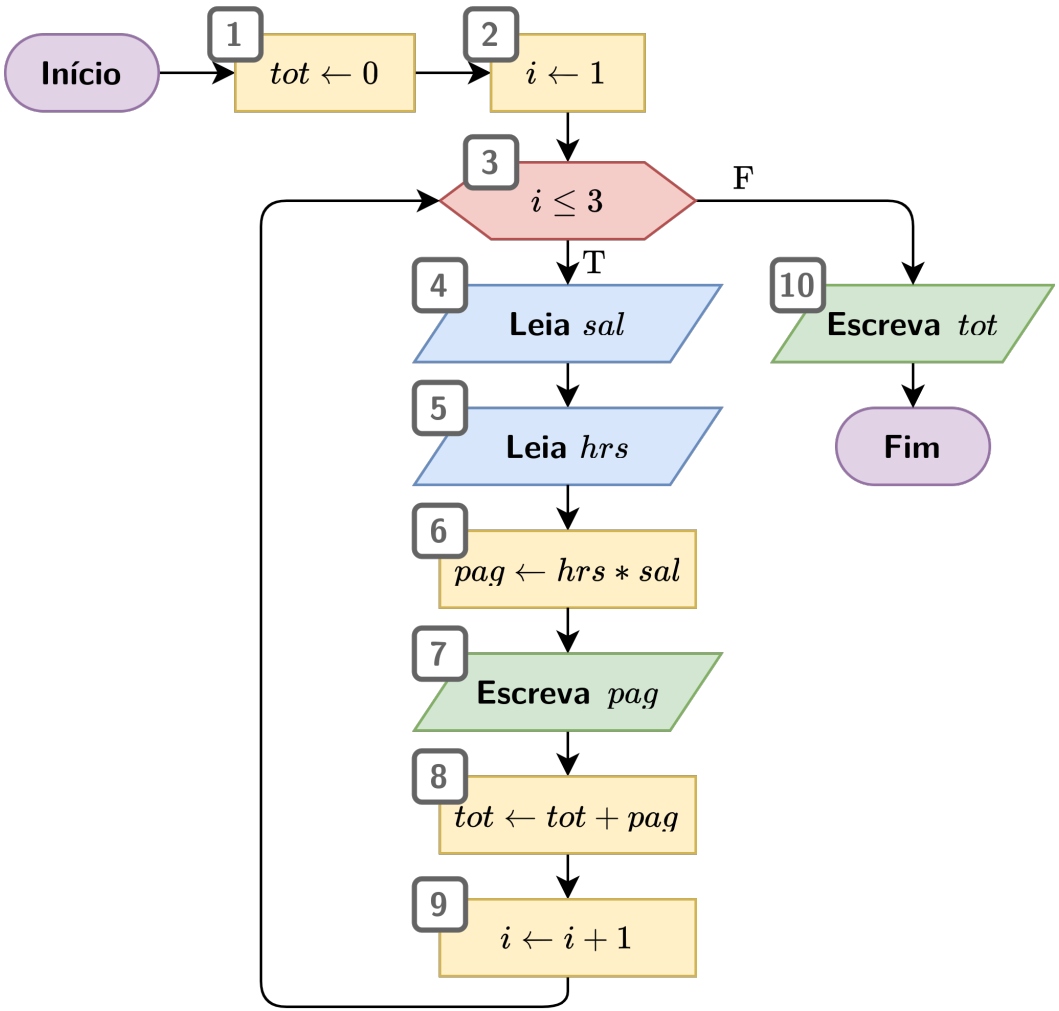
Instr	sal	hrs	pag	i	tot
6	60	7	420	2	400
7	60	7	420	2	400
8	60	7	420	2	820
9	60	7	420	3	820
3	60	7	420	3	820
4	50	7	420	3	820
5	50	7.5	420	3	820



Entrada: ~~50 8 60 7 50 7.5~~ Saída: 400 420

Instr	sal	hrs	pag	i	tot
Início	?	?	?	?	?
1	?	?	?	?	0
2	?	?	?	1	0
3	?	?	?	1	0
4	50	?	?	1	0
5	50	8	?	1	0
6	50	8	400	1	0
7	50	8	400	1	0
8	50	8	400	1	400
9	50	8	400	2	400
3	50	8	400	2	400
4	60	8	400	2	400
5	60	7	400	2	400

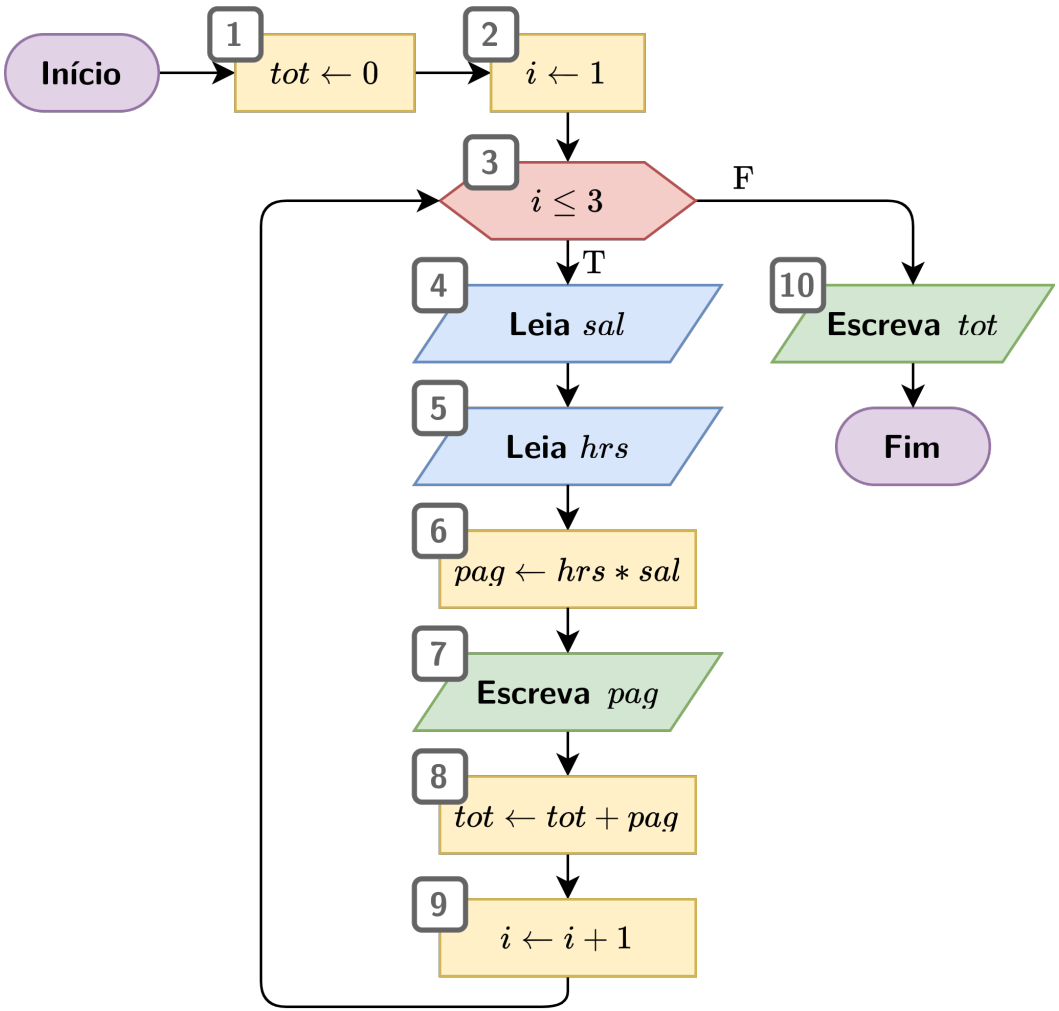
Instr	sal	hrs	pag	i	tot
6	60	7	420	2	400
7	60	7	420	2	400
8	60	7	420	2	820
9	60	7	420	3	820
3	60	7	420	3	820
4	50	7	420	3	820
5	50	7.5	420	3	820
6	50	7.5	375	3	820



Entrada: ~~50 8 60 7 50 7.5~~ Saída: 400 420 375

Instr	sal	hrs	pag	i	tot
Início	?	?	?	?	?
1	?	?	?	?	0
2	?	?	?	1	0
3	?	?	?	1	0
4	50	?	?	1	0
5	50	8	?	1	0
6	50	8	400	1	0
7	50	8	400	1	0
8	50	8	400	1	400
9	50	8	400	2	400
3	50	8	400	2	400
4	60	8	400	2	400
5	60	7	400	2	400

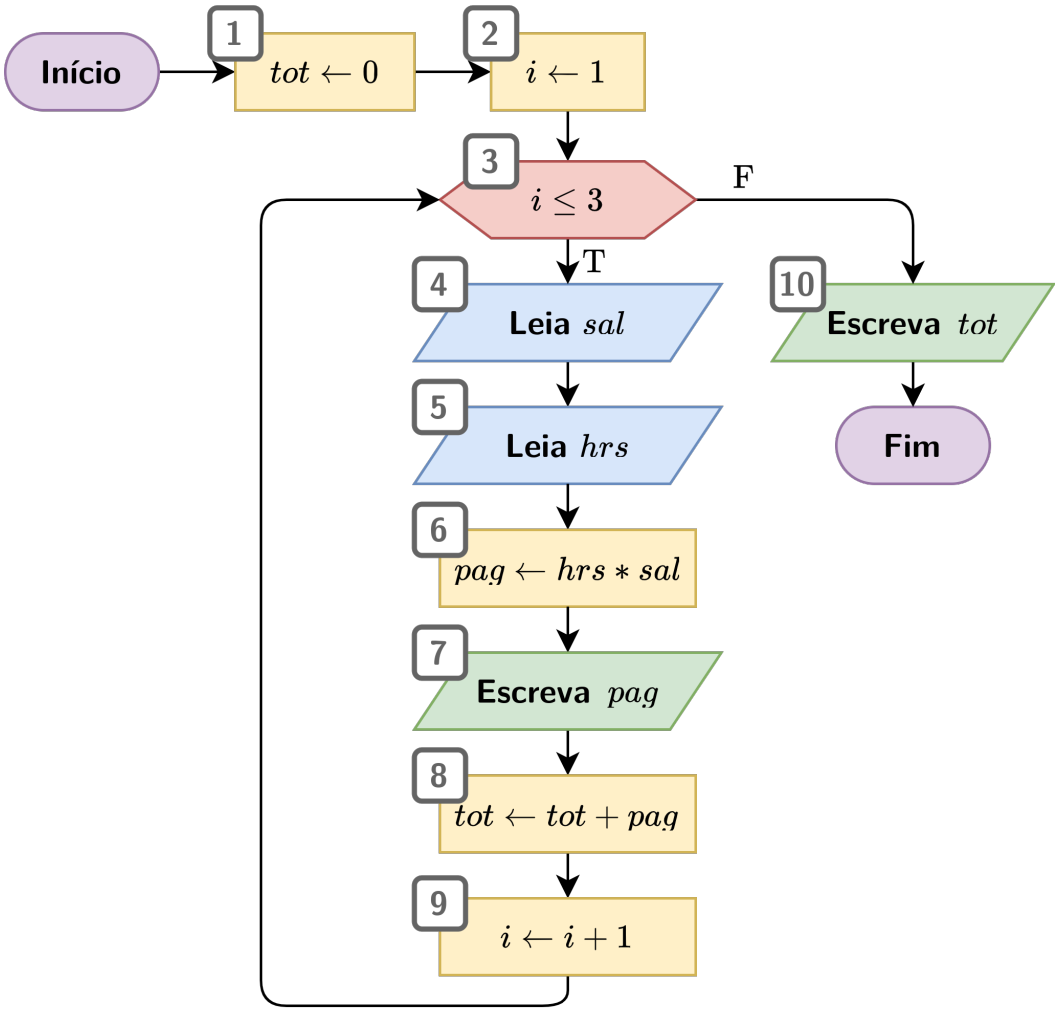
Instr	sal	hrs	pag	i	tot
6	60	7	420	2	400
7	60	7	420	2	400
8	60	7	420	2	820
9	60	7	420	3	820
3	60	7	420	3	820
4	50	7	420	3	820
5	50	7.5	420	3	820
6	50	7.5	375	3	820
7	50	7.5	375	3	820



Entrada: ~~50 8 60 7 50 7.5~~ Saída: 400 420 375

Instr	*sal*	*hrs*	*pag*	*i*	*tot*
Início	?	?	?	?	?
1	?	?	?	?	0
2	?	?	?	1	0
3	?	?	?	1	0
4	50	?	?	1	0
5	50	8	?	1	0
6	50	8	400	1	0
7	50	8	400	1	0
8	50	8	400	1	400
9	50	8	400	2	400
3	50	8	400	2	400
4	60	8	400	2	400
5	60	7	400	2	400

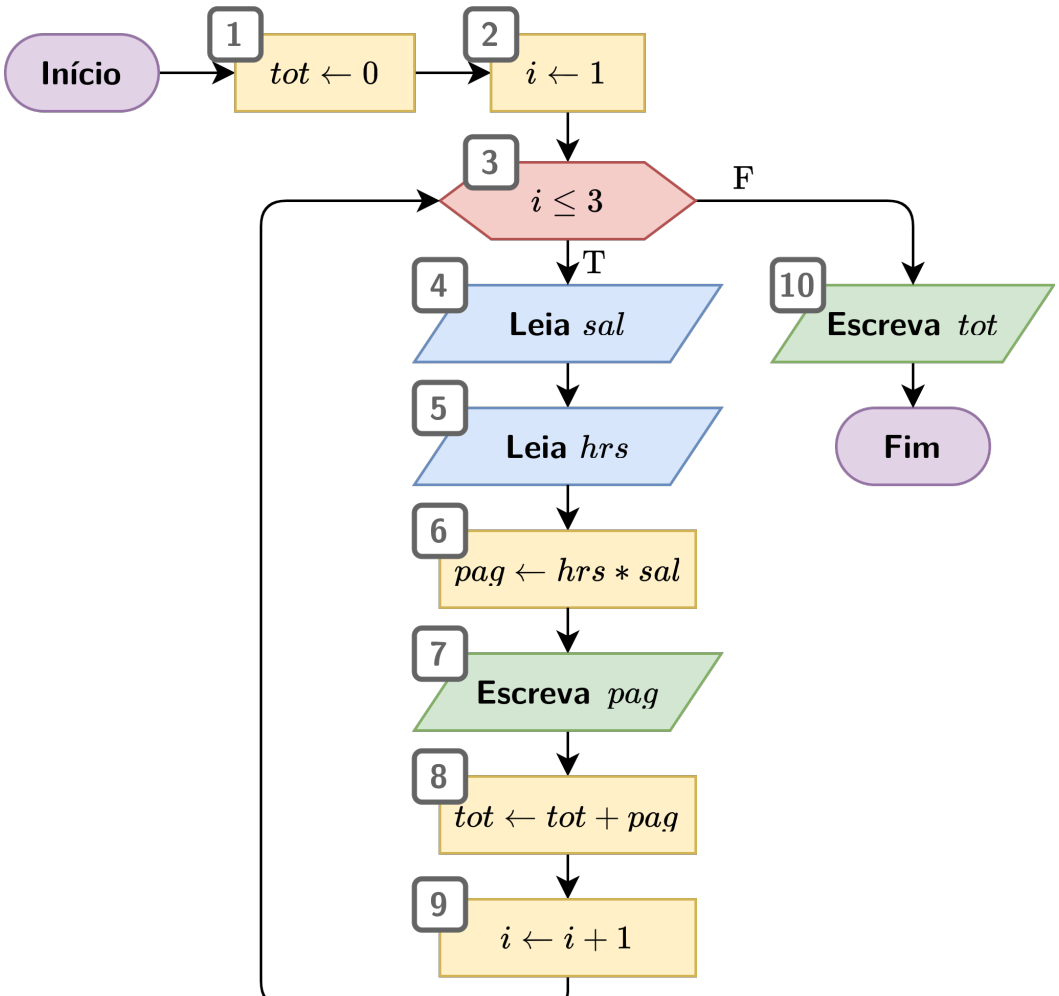
Instr	*sal*	*hrs*	*pag*	*i*	*tot*
6	60	7	420	2	400
7	60	7	420	2	400
8	60	7	420	2	820
9	60	7	420	3	820
3	60	7	420	3	820
4	50	7	420	3	820
5	50	7.5	420	3	820
6	50	7.5	375	3	820
7	50	7.5	375	3	820
8	50	7.5	375	3	1195



Entrada: ~~50 8 60 7 50 7.5~~ Saída: 400 420 375

Instr	sal	hrs	pag	i	tot
Início	?	?	?	?	?
1	?	?	?	?	0
2	?	?	?	1	0
3	?	?	?	1	0
4	50	?	?	1	0
5	50	8	?	1	0
6	50	8	400	1	0
7	50	8	400	1	0
8	50	8	400	1	400
9	50	8	400	2	400
3	50	8	400	2	400
4	60	8	400	2	400
5	60	7	400	2	400

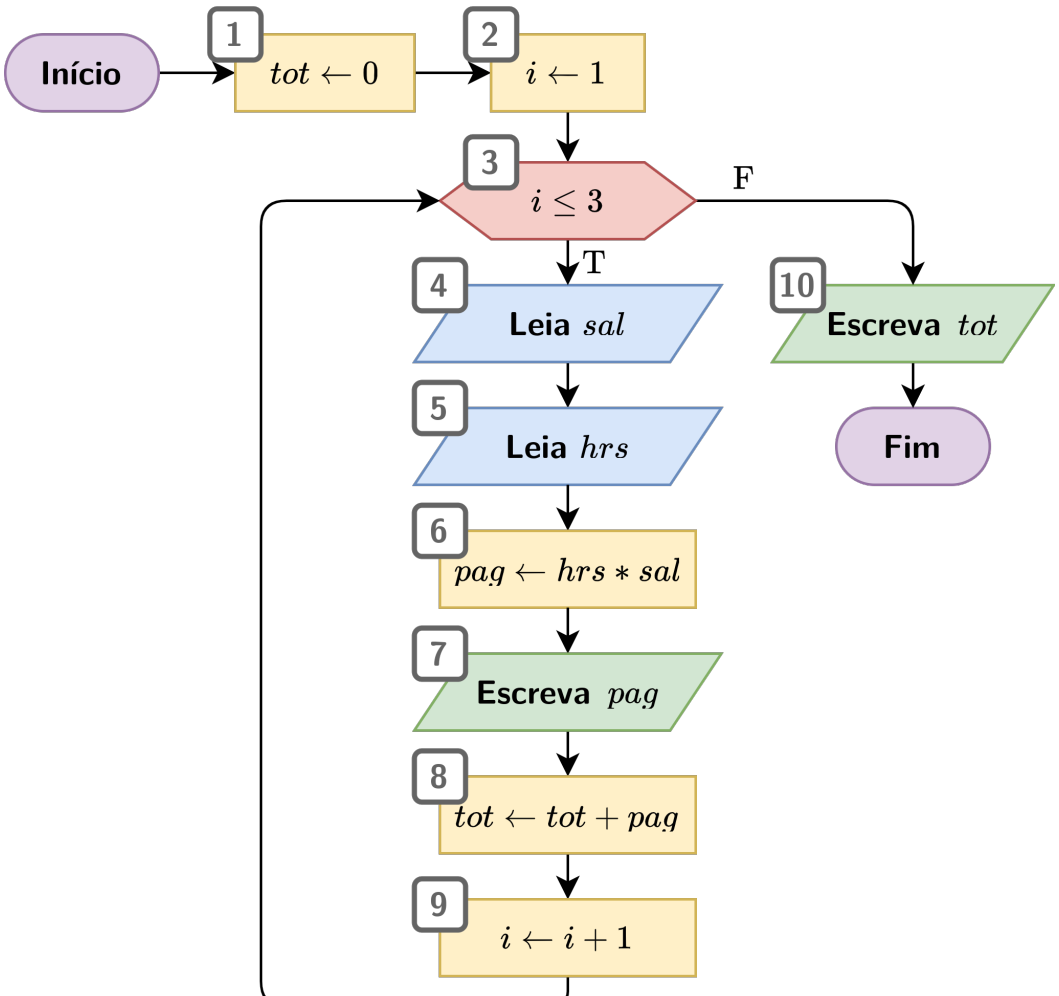
Instr	sal	hrs	pag	i	tot
6	60	7	420	2	400
7	60	7	420	2	400
8	60	7	420	2	820
9	60	7	420	3	820
3	60	7	420	3	820
4	50	7	420	3	820
5	50	7.5	420	3	820
6	50	7.5	375	3	820
7	50	7.5	375	3	820
8	50	7.5	375	3	1195
9	50	7.5	375	4	1195



Entrada: ~~50 8 60 7 50 7.5~~ Saída: 400 420 375

Instr	sal	hrs	pag	i	tot
Início	?	?	?	?	?
1	?	?	?	?	0
2	?	?	?	1	0
3	?	?	?	1	0
4	50	?	?	1	0
5	50	8	?	1	0
6	50	8	400	1	0
7	50	8	400	1	0
8	50	8	400	1	400
9	50	8	400	2	400
3	50	8	400	2	400
4	60	8	400	2	400
5	60	7	400	2	400

Instr	sal	hrs	pag	i	tot
6	60	7	420	2	400
7	60	7	420	2	400
8	60	7	420	2	820
9	60	7	420	3	820
3	60	7	420	3	820
4	50	7	420	3	820
5	50	7.5	420	3	820
6	50	7.5	375	3	820
7	50	7.5	375	3	820
8	50	7.5	375	3	1195
9	50	7.5	375	4	1195
3	50	7.5	375	4	1195



Entrada: ~~50 8 60 7 50 7.5~~ Saída: 400 420 375 1195

Instr	sal	hrs	pag	i	tot
Início	?	?	?	?	?
1	?	?	?	?	0
2	?	?	?	1	0
3	?	?	?	1	0
4	50	?	?	1	0
5	50	8	?	1	0
6	50	8	400	1	0
7	50	8	400	1	0
8	50	8	400	1	400
9	50	8	400	2	400
3	50	8	400	2	400
4	60	8	400	2	400
5	60	7	400	2	400

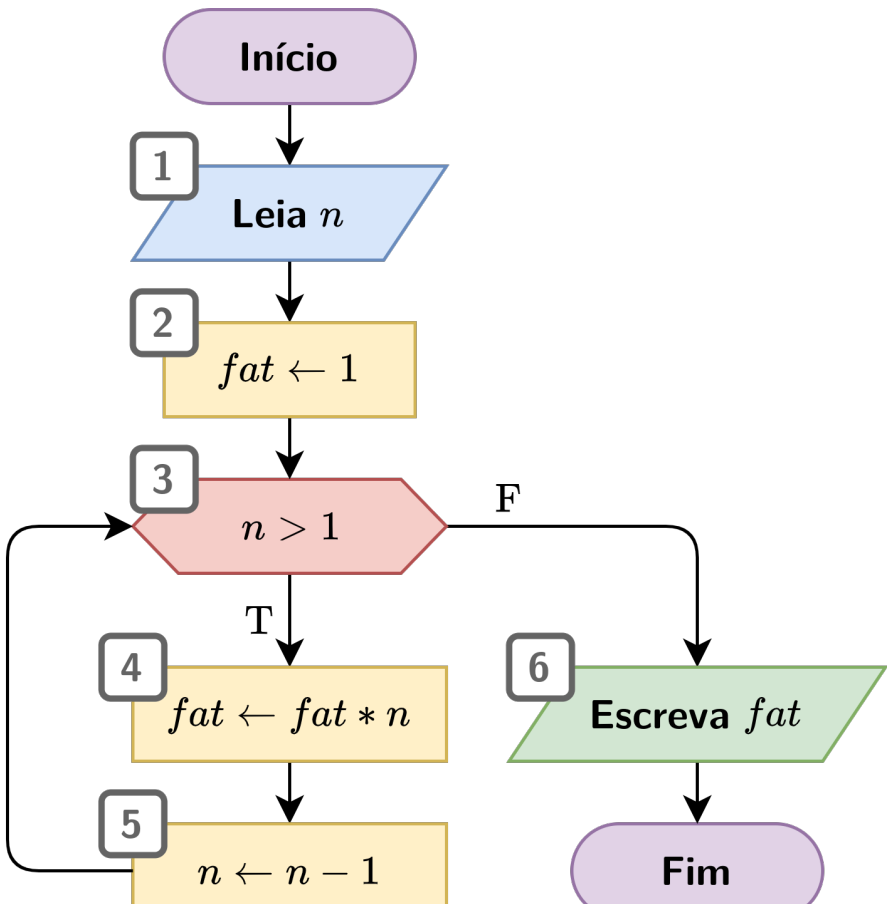
Instr	sal	hrs	pag	i	tot
6	60	7	420	2	400
7	60	7	420	2	400
8	60	7	420	2	820
9	60	7	420	3	820
3	60	7	420	3	820
4	50	7	420	3	820
5	50	7.5	420	3	820
6	50	7.5	375	3	820
7	50	7.5	375	3	820
8	50	7.5	375	3	1195
9	50	7.5	375	4	1195
3	50	7.5	375	4	1195
10	50	7.5	375	4	1195

Exemplo: Fatorial

Elabore um algoritmo que lê um inteiro não-negativo e escreve o fatorial desse número. Lembre-se:

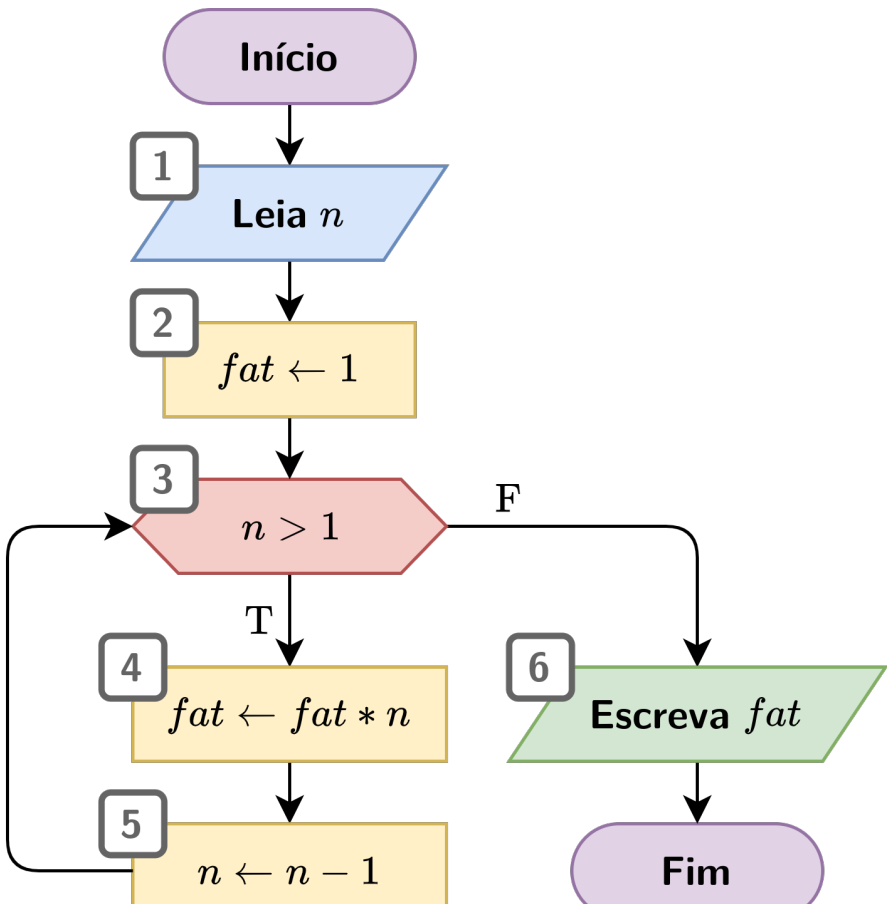
$$n! = n \times (n-1) \times (n-2) \times \cdots \times 2 \times 1.$$

Além disso, $0! = 1$, por definição.



Variáveis
n: inteiro
fat: inteiro

```
1 Início
2     Leia n
3     fat ← 1
4     Enquanto n > 1 faça
5         fat ← fat * n
6         n ← n − 1
7     FimEnquanto
8     Escreva fat
9 Fim
```

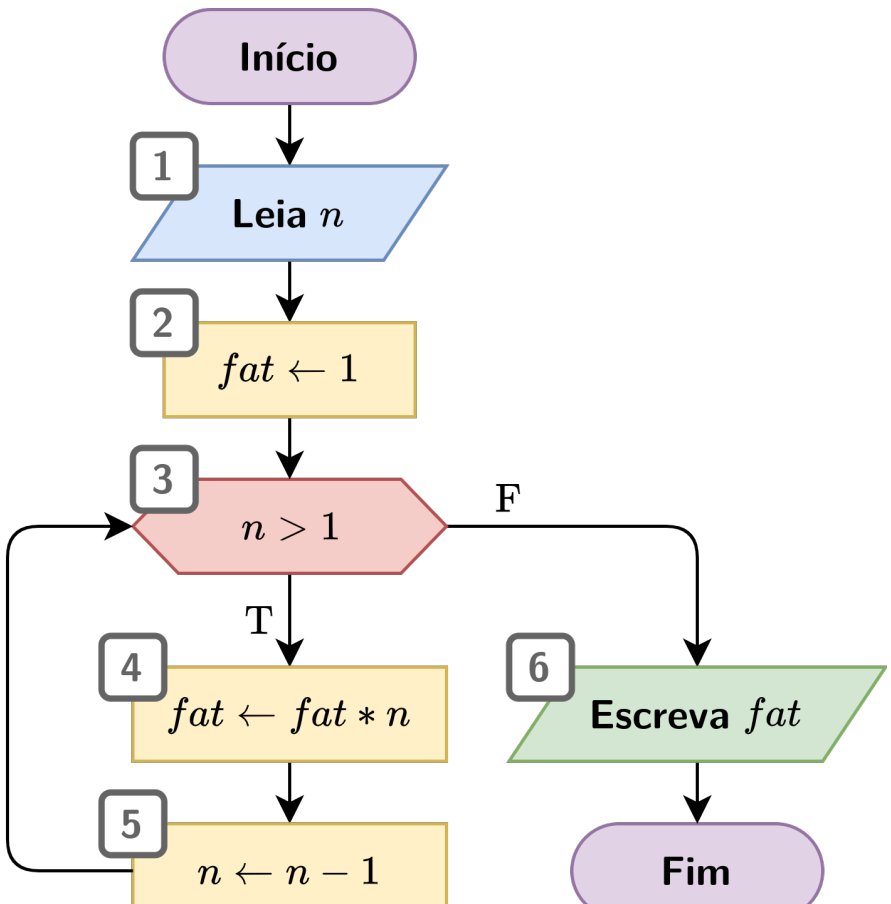


Entrada: 4

Instr	n	fat
Início	?	?

Instr	n	fat

Saída:

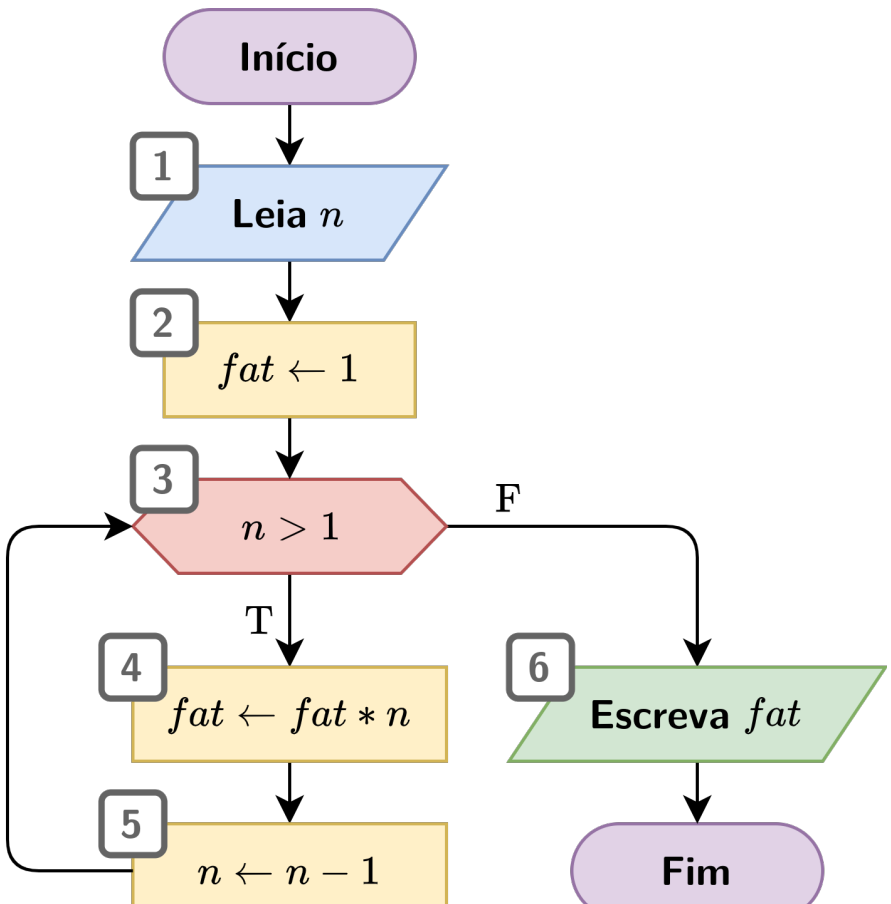


Entrada: ~~4~~

Instr	n	fat
Início	?	?
1	4	?

Instr	n	fat

Saída:

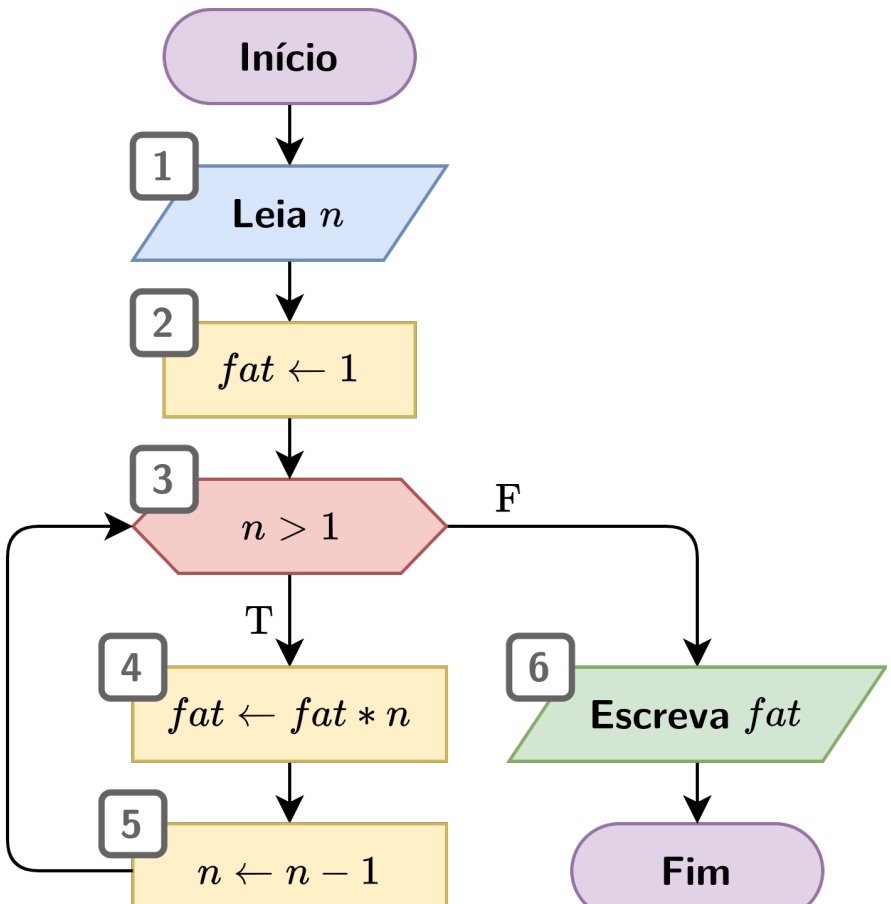


Entrada: 4

Instr	n	fat
Início	?	?
1	4	?
2	4	1

Instr	n	fat

Saída:

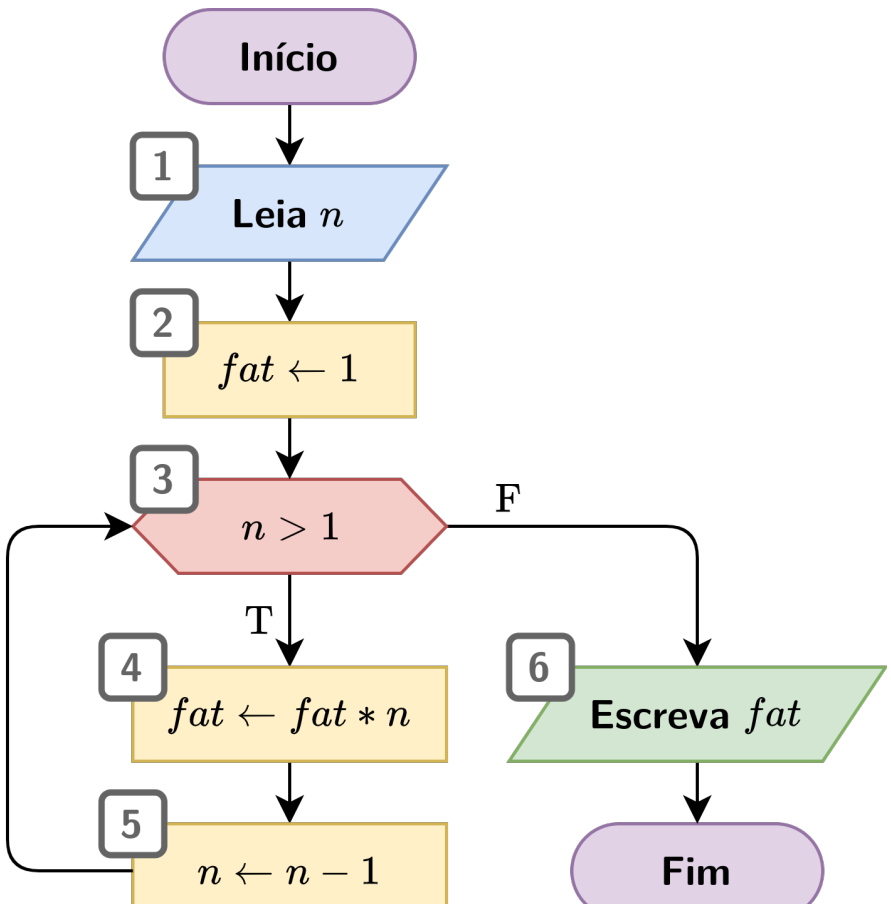


Entrada: 4

Instr	n	fat
Início	?	?
1	4	?
2	4	1
3	4	1

Instr	n	fat

Saída:

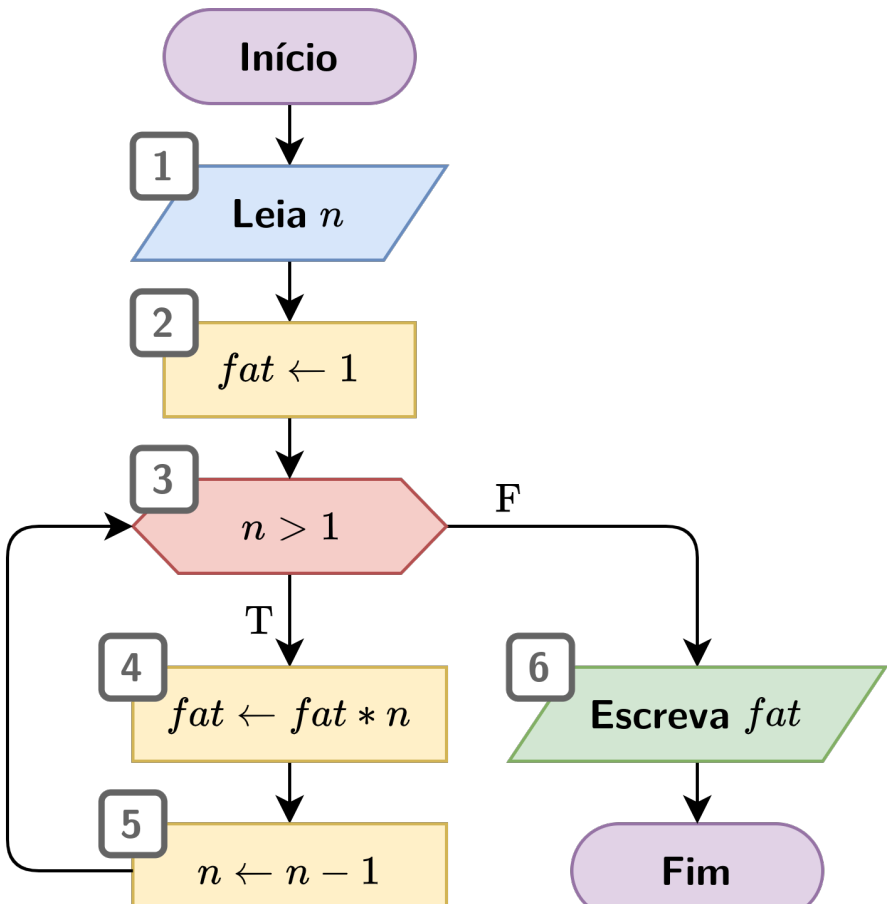


Entrada: 4

Instr	n	fat
Início	?	?
1	4	?
2	4	1
3	4	1
4	4	4

Instr	n	fat

Saída:

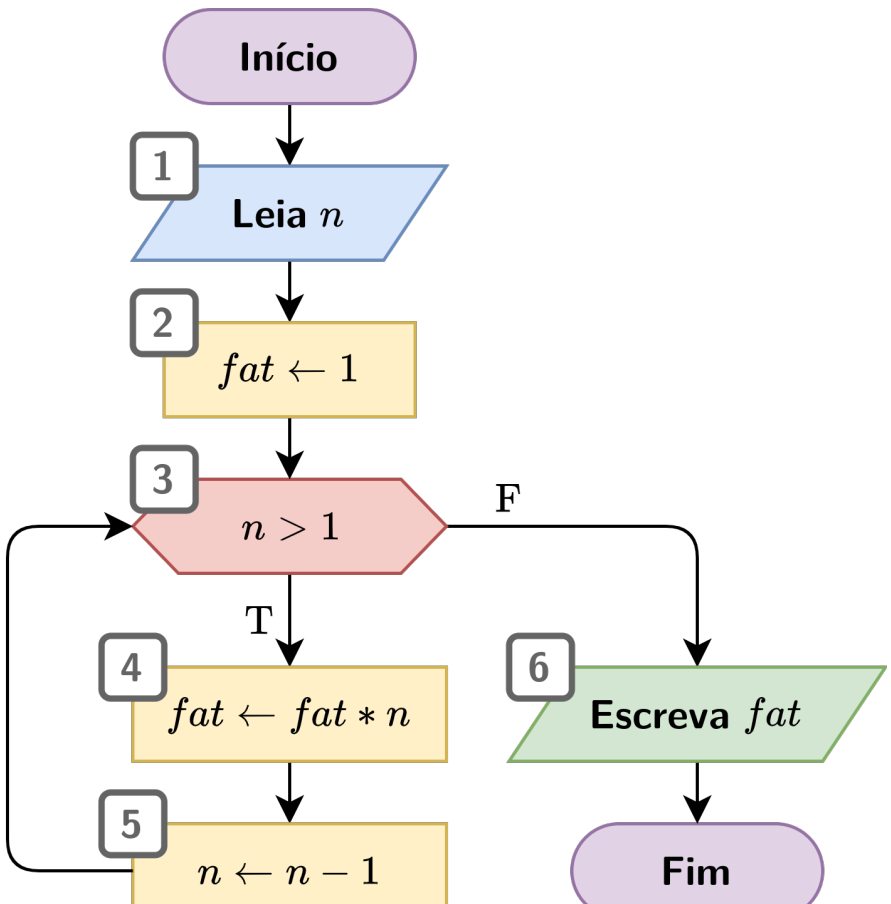


Entrada: 4

Instr	n	fat
Início	?	?
1	4	?
2	4	1
3	4	1
4	4	4
5	3	4

Instr	n	fat

Saída:

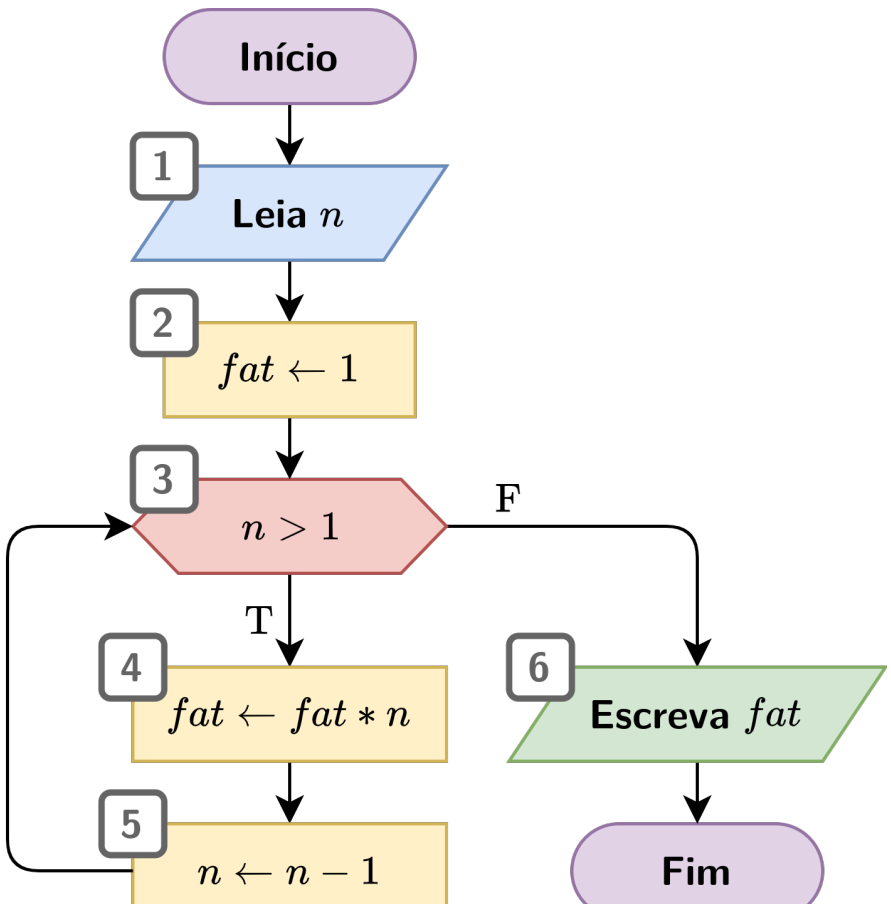


Entrada: ~~4~~

Instr	n	fat
Início	?	?
1	4	?
2	4	1
3	4	1
4	4	4
5	3	4
3	3	4

Instr	n	fat

Saída:

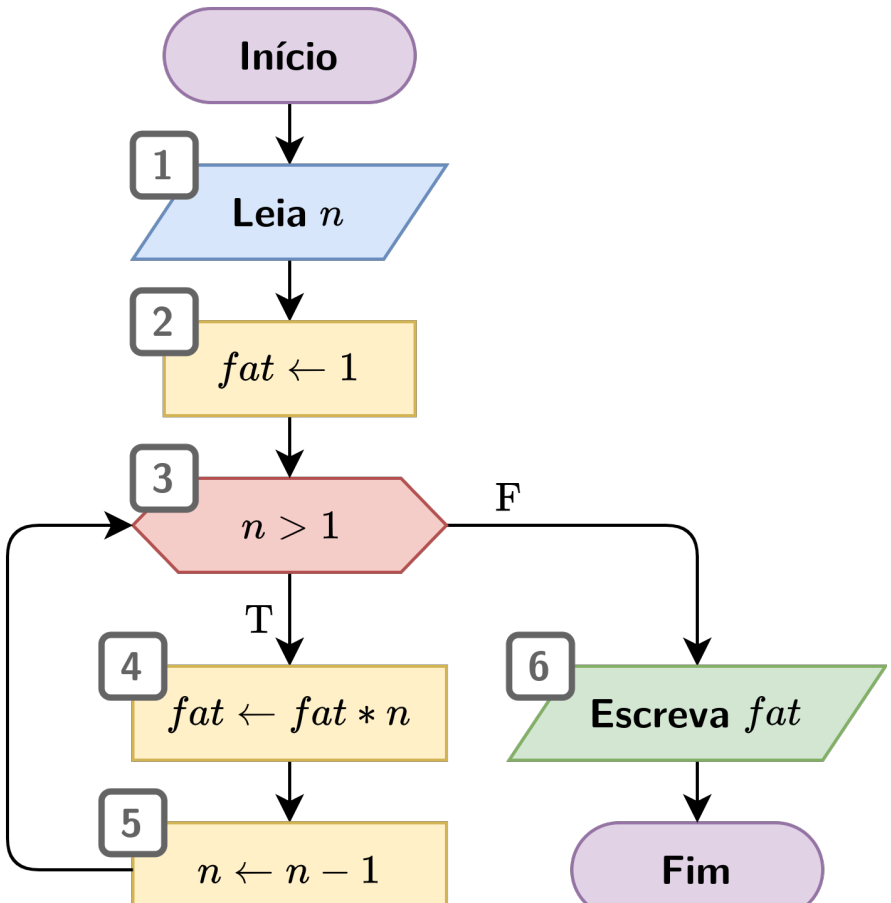


Entrada: 4

Instr	n	fat
Início	?	?
1	4	?
2	4	1
3	4	1
4	4	4
5	3	4
3	3	4
4	3	12

Saída:

Instr	n	fat

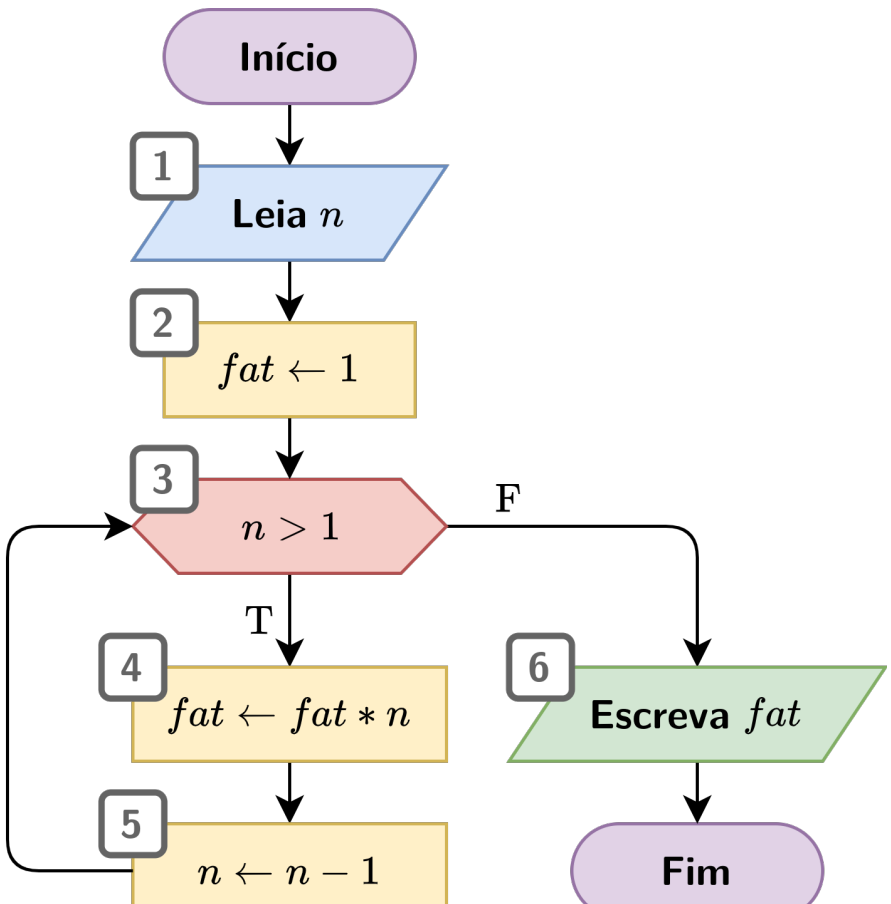


Entrada: 4

Instr	n	fat
Início	?	?
1	4	?
2	4	1
3	4	1
4	4	4
5	3	4
3	3	4
4	3	12

Instr	n	fat
5	2	12

Saída:

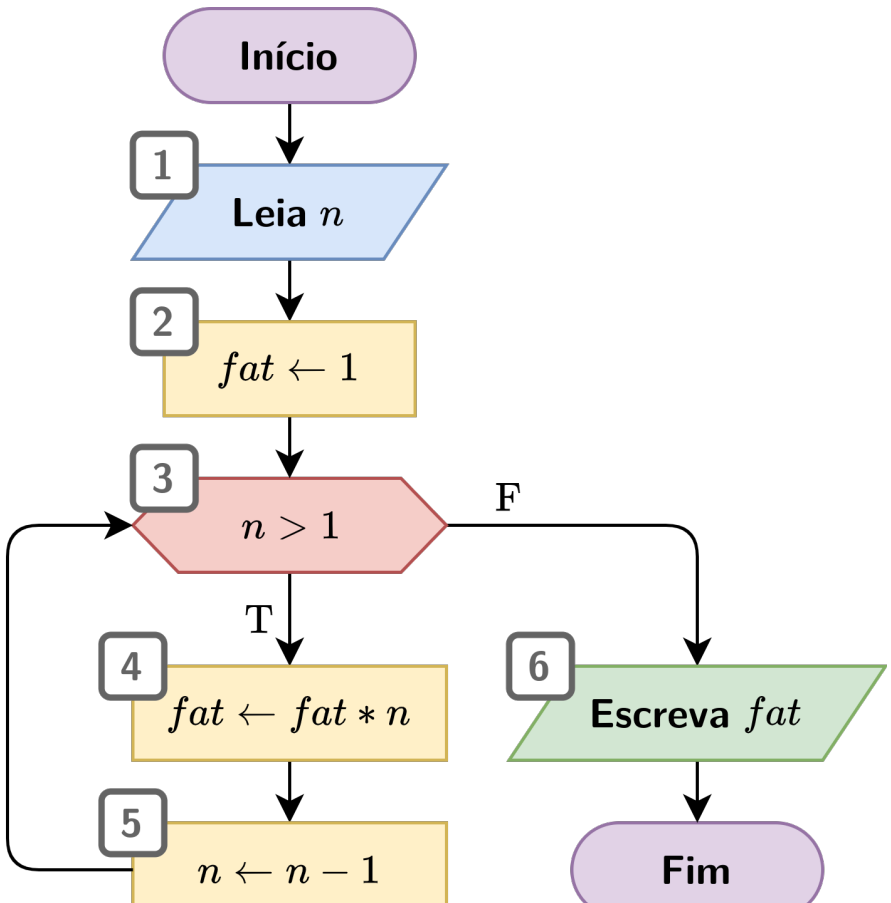


Entrada: 4

Instr	n	fat
Início	?	?
1	4	?
2	4	1
3	4	1
4	4	4
5	3	4
3	3	4
4	3	12

Instr	n	fat
5	2	12
3	2	12

Saída:

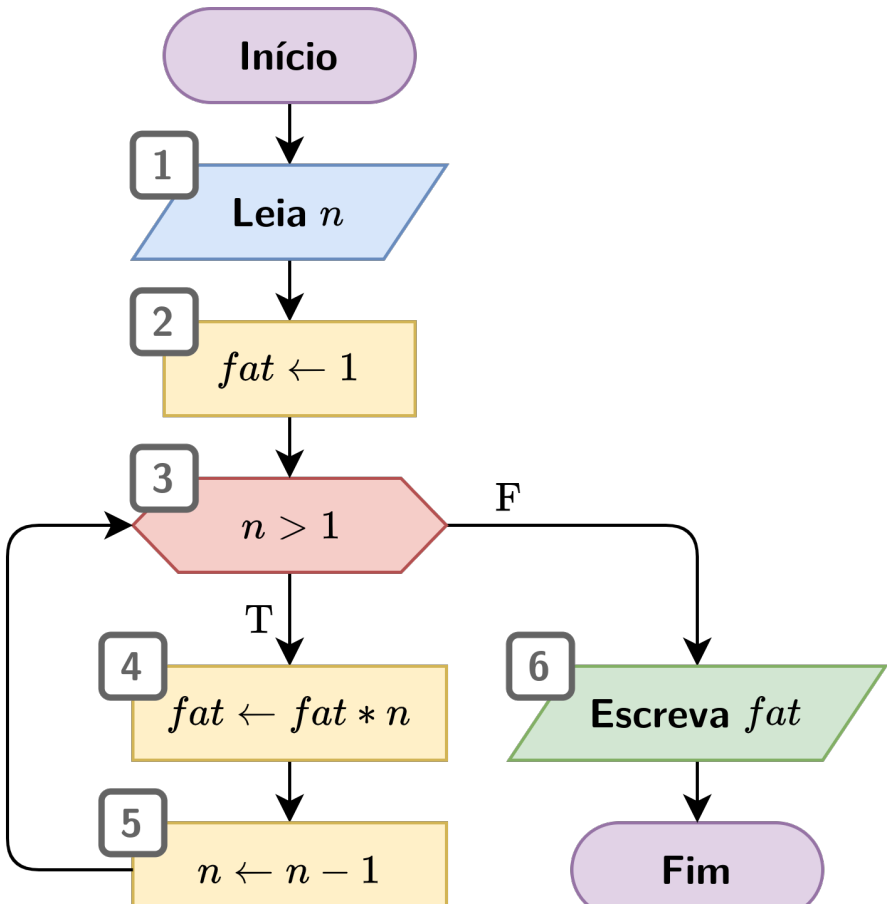


Entrada: 4

Instr	n	fat
Início	?	?
1	4	?
2	4	1
3	4	1
4	4	4
5	3	4
3	3	4
4	3	12

Instr	n	fat
5	2	12
3	2	12
4	2	24

Saída:

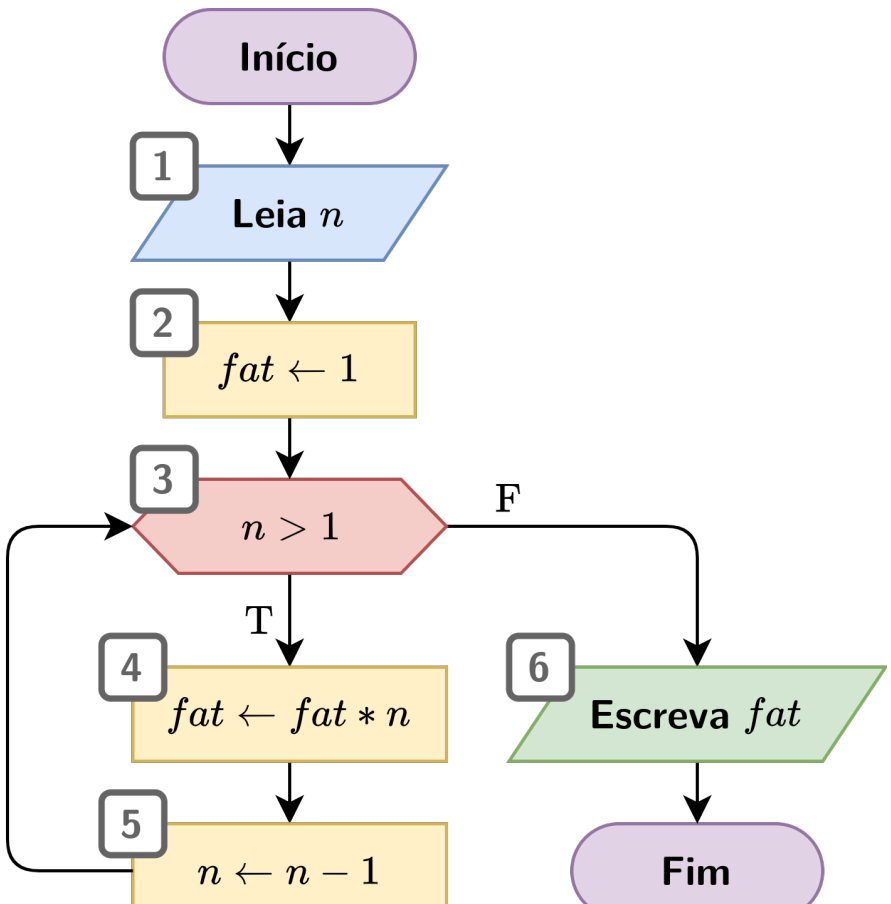


Entrada: 4

Instr	n	fat
Início	?	?
1	4	?
2	4	1
3	4	1
4	4	4
5	3	4
3	3	4
4	3	12

Instr	n	fat
5	2	12
3	2	12
4	2	24
5	1	24

Saída:

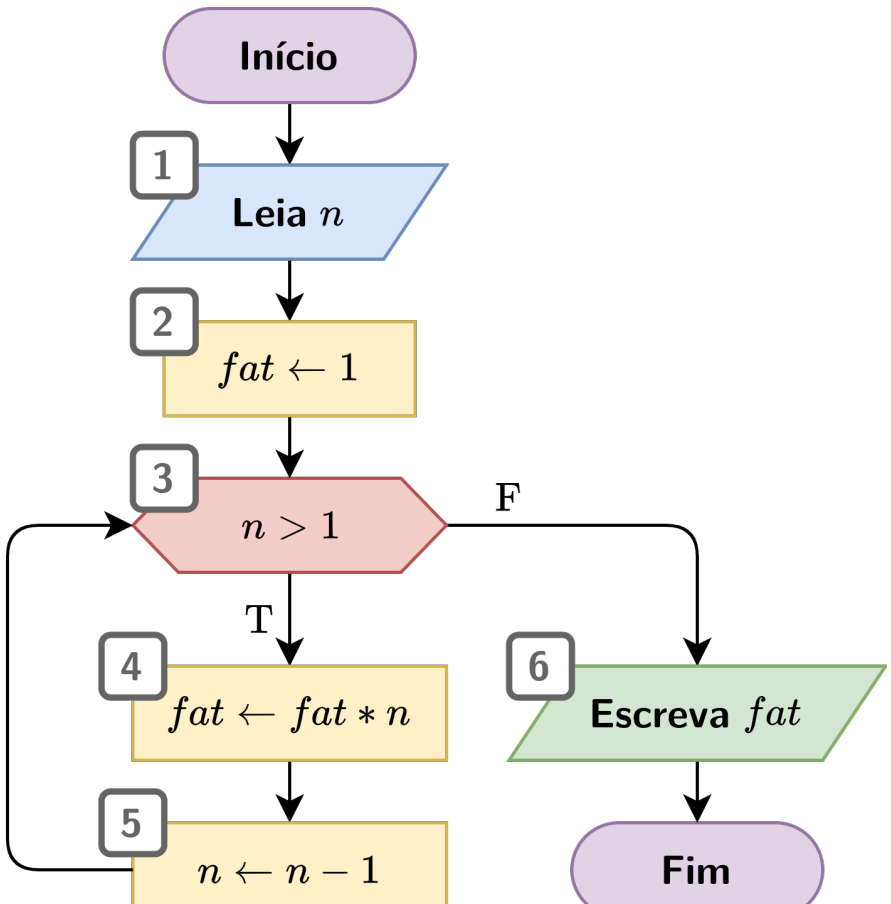


Entrada: ~~4~~

Instr	n	fat
Início	?	?
1	4	?
2	4	1
3	4	1
4	4	4
5	3	4
3	3	4
4	3	12

Instr	n	fat
5	2	12
3	2	12
4	2	24
5	1	24
3	1	24

Saída:



Entrada: 4

Instr	n	fat
Início	?	?
1	4	?
2	4	1
3	4	1
4	4	4
5	3	4
3	3	4
4	3	12

Instr	n	fat
5	2	12
3	2	12
4	2	24
5	1	24
3	1	24
6	1	24

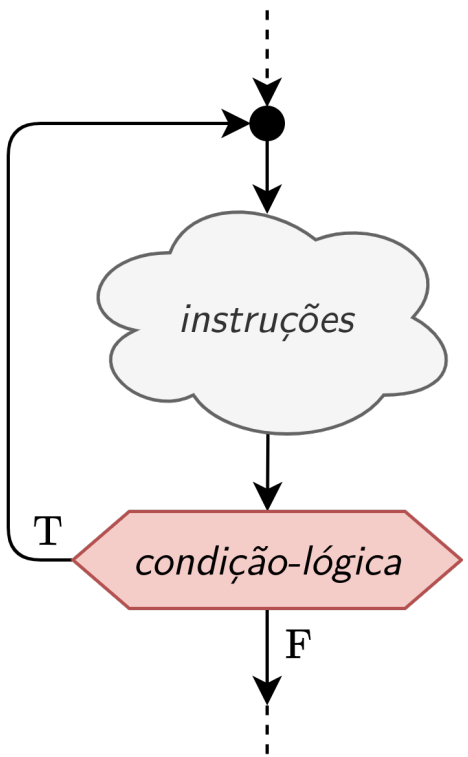
Saída: 24

Para pensar:

1. O algoritmo funciona para os casos $n = 1$ e $n = 0$?
2. Como fazer para que o valor de n não seja perdido?
3. É possível fazer uma versão em que o produto seja feito da forma $1 \times 2 \times 3 \times \cdots \times n$?
4. O que acontece se o usuário digitar um número negativo ou um número não inteiro?

Comando Faça-Enquanto

Faça
    instruções
Enquanto *condição-lógica*

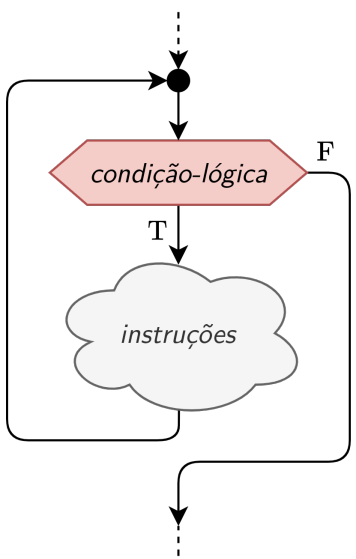


Enquanto vs. Faça-Enquanto

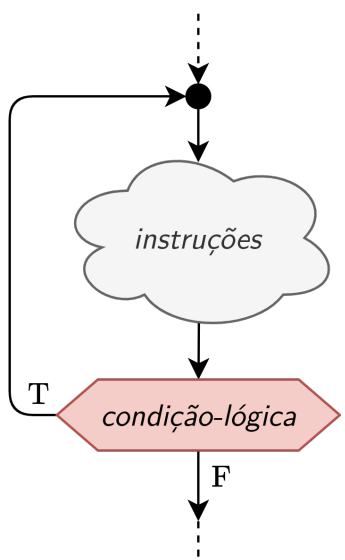
No comando **Enquanto**, é possível que o corpo nunca seja executado, pois a condição é testada logo no início.

Já no comando **Faça-Enquanto**, o corpo é executado *pelo menos uma vez*, pois a condição é testada no final.

Enquanto *condição-lógica* **faça**
    instruções
FimEnquanto



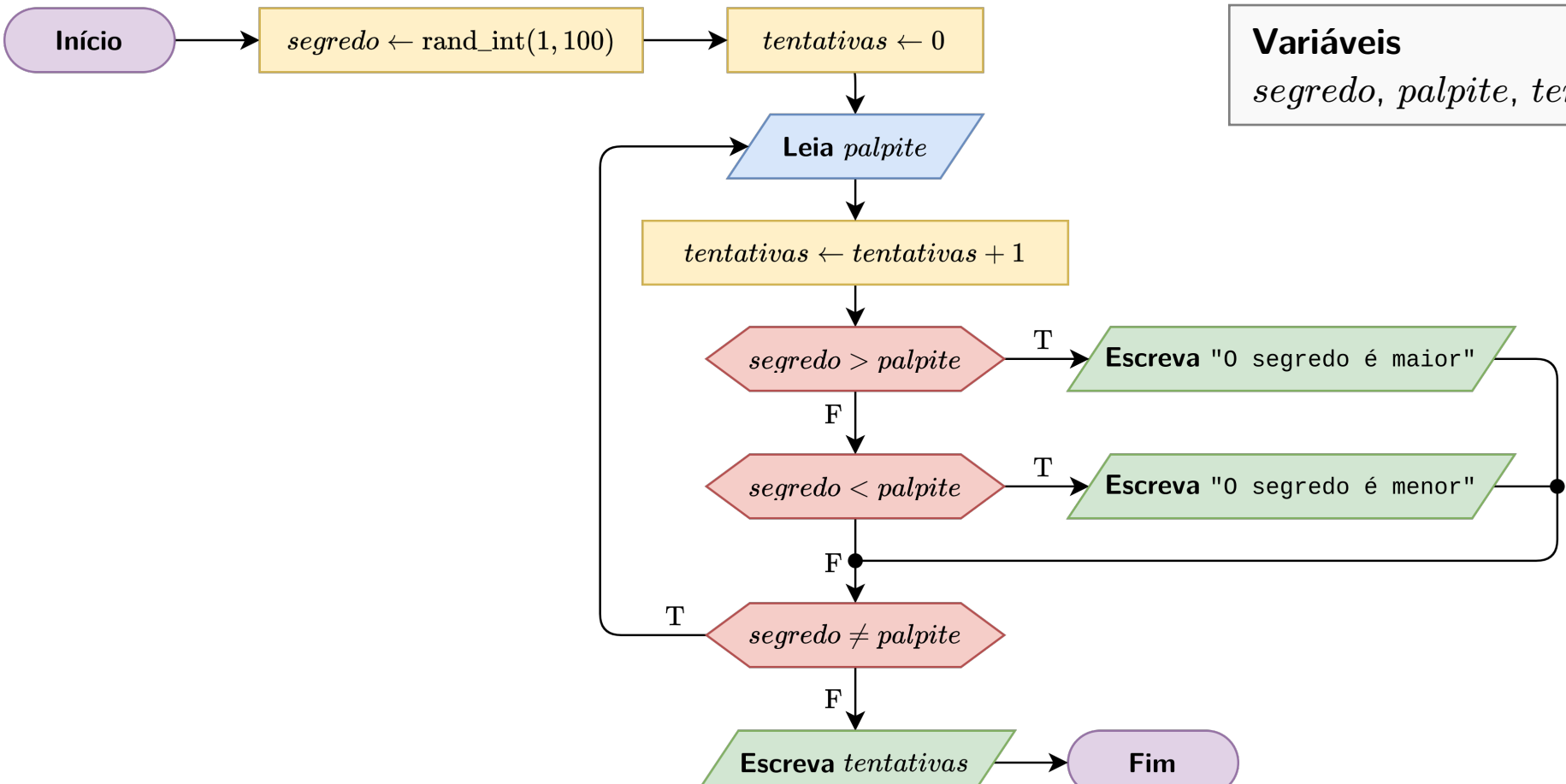
Faça
    instruções
Enquanto *condição-lógica*



Número secreto

Implemente um algoritmo que simula um **jogo de adivinhação** de um **número secreto**.

- O jogo deve gerar um número aleatório entre 1 e 100, e solicitar ao usuário que digite um palpite.
- Se o usuário acertar o número, o jogo deve informar o número de tentativas que foram necessárias para acertar.
- Caso contrário, o jogo deve informar se o número secreto é *maior* ou *menor* que o palpite, e solicitar um novo palpite.
- O jogo deve continuar solicitando novos palpites até que o usuário acerte o número secreto.


Início
segredo ← rand_int(1, 100)
tentativas ← 0
Leia palpite
tentativas ← tentativas + 1
segredo > palpite
T
Escreva "O segredo é maior"
F
segredo < palpite
T
Escreva "O segredo é menor"
F
T
segredo ≠ palpite
F
Escreva tentativas
Fim
Variáveis
segredo, palpite, tentativas: inteiro

Variáveis
$segredo$, $palpite$, $tentativas$: inteiro

```
1  Início
2      segredo ← rand_int(1, 100)
3      tentativas ← 0
4      Faça
5          Leia palpite
6          tentativas ← tentativas + 1
7          Se segredo > palpite então
8              Escreva "O segredo é maior"
9          SenãoSe segredo < palpite
10             Escreva "O segredo é menor"
11         FimSe
12     Enquanto segredo ≠ palpite
13     Escreva tentativas
14 Fim
```

Exercícios em sala de aula

1. Elabore um fluxograma e um pseudocódigo para um algoritmo que LÊ um número inteiro de referência, seguido de mais dez números inteiros e ESCREVE a quantidade de números maiores que o número de referência. Por exemplo, se a entrada for 5, 3, 1, 4, 1, 5, 9, 2, 6, 5, 3, a saída deve ser 2. Utilize o comando **Enquanto**.

2. (a) Elabore um fluxograma e um pseudocódigo para um algoritmo que LÊ um inteiro positivo n, seguido de n números inteiros (assuma todos positivos) e ESCREVE o maior número lido. Utilize **Enquanto**. Em seguida, efetue um teste de mesa com a entrada 5, 7, 2, 15, 21, 15; a saída deve ser 21.

   (b) Suponha agora não será informado o número de valores a serem lidos. No lugar, a leitura dos números é encerrada quando o usuário entrar com 0 (zero). Utilize **Faça-Enquanto**. Em seguida, efetue um teste de mesa com a entrada 7, 2, 15, 21, 15, 0; a saída deve ser 21.

3. Elabore um fluxograma e um pseudocódigo para um algoritmo que LÊ um inteiro positivo e ESCREVE a quantidade de dígitos do número lido. Em seguida, efetue um teste de mesa com a entrada 1729; a saída deve ser 4. *Dica:* Utilize o operador de divisão inteira.