

motor E (1)(3)	Fila	Volume	Fila < 2	Volume < 3	Pontaria
motor E (1)(3) ← 0	0	0	0 < 2	0	0
motor E (1)(3) ← 1	1	1	1 < 2	1	1 < 3
motor E (1)(3) ← 2	2	2	2 < 2	2	2 < 3
motor E (1)(3) ← 3	3	3	3 < 3	3	3 < 3
motor E (1)(3) ← 0	0	0	0 < 3	0	0
motor E (1)(3) ← 1	1	1	1 < 3	1	1
motor E (1)(3) ← 2	2	2	2 < 3	2	2
motor E (1)(3) ← 3	3	3	3 < 3	3	3

1. fila 2. volume

motor (fila 1 volume), tipo de sorte (domínio)

↓
variável

Pontaria		Pontaria	
0 < 2	0	0	0
1 < 2	1	1	1
2 < 2	2	2	2
3 < 2	3	3	3
0 < 3	0	0	0
1 < 3	1	1	1
2 < 3	2	2	2
3 < 3	3	3	3

↓
Variável

file	0	1	2	3
0 < 2	0	1	2	3
1 < 2	1	2	3	0
2 < 2	2	3	0	1
3 < 2	3	0	1	2

L... linea

2.. m_{tro}E (2)(3), i [t-n])

3.. fila, i [0-n)

4.. columna, i [0-n]

5.. columna, i [0-n]

6.. Pasa (fila < 0, fila < 1,) loop

7.. Pasa (columna < 0, columna < 3,) loop

8.. Imprimir "Imprime valor para " + fila + " " + columna

9.. Leer valor

10.. Si columna >= 4 and columna <= 10 entonces

m_{tro}E [fila][columna] ← valor~~SW~~

11.. Llene sentencia

m_{tro}F [fila][columna] ← L

14.. fin si

15.. fin para

16.. fin para

17.. Pasa (fila < 0, fila < 2,) loop

18.. Pasa (columna < 0, columna < 3,) loop

19.. Imprimir fila + " " + columna + " valor: " + m_{tro}F [fila][columna]

20.. Fin para

21.. fin para

22..

fib_n [1, 2] =
 0 1 2
 0 1 0 2
 0 1 0 2 3
 1 0 1 2 3

fib_n Sum
 0 0 0
 1 1 1
 2 2 3
 3 3 6

fib_n Sum
 0 0 0
 1 1 1
 2 2 3
 3 3 6

fib_n Sum
 0 0 0
 1 1 1
 2 2 3
 3 3 6

fib_n Sum
 0 0 0
 1 1 1
 2 2 3
 3 3 6

fib_n (1, 3)
 0 1 2
 0 1 0 2 3
 1 0 1 2 3

fib_n
 0 0 0
 1 1 1
 2 2 3
 3 3 6
 4 4 10
 5 5 15

fib_n Sum
 0 0 0
 1 1 1
 2 2 3
 3 3 6
 4 4 10
 5 5 15

fib_n Sum
 0 0 0
 1 1 1
 2 2 3
 3 3 6
 4 4 10
 5 5 15

~~fib_n~~

~~fib_n~~