



LÓGICA DE PROGRAMAÇÃO EXERCÍCIOS

+55 (21) 99461-8818

@explicadoresnet

www.explicadores.net.br



```

var
    S, C: Inteiro
inicio
    S <- 0
    C <- 1
    Enquanto (C <= 10) faça
        S <- S + C
        C <- C + 4
    FimEnquanto
    Escreva(S)
finalgoritmo

```

REFER.ÇÃO
 COM CONDIÇÃO
 NO INÍCIO
 DO
 LOOPING

S	C
0	1
4	5
8	9
12	13
15	17

15

1 2 / 0
0

ESTRUTURA DE
REPETIÇÃO COM
CONDICÃO NO
FINAL DO
LOOPING

```

var
  X, Y, Z: Inteiro
inicio
  X <- 1
  Y <- 0
  Repita
    Se (X % 2 = 0) então
      Z <- 5
    senao
      Z <- 8
    FimSe
    Y <- Y + Z
    X <- X + 1
  Ate (X > 4)
  Escreva (Y)
finalgoritmo
  
```

MÓDULO DA DIVISÃO
MOD (RESTO DA
DIVISÃO
INFERNA)

X	Y	Z
1	0	8
2	8	5
3	8	5
4	24	5
5	26	5

26



ESTRUTURA DE
REPETIÇÃO COM
VARIÁVEL DE
CONTROLE.

```

var
  c, i, x: Inteiro
inicio
  c <- 0
  i <- 5
  Para x <- 0 ate 20 passo 1 5 faça
    Se (x < i + 10) entao
      Escreva(x)
    senao
      Escreva(x+i)
    FimSe
  FimPara
finalgoritmo
  
```

c	i	x
0	5	0
		1
		2
		3
		4
		5
		10
		15
		20

0 5 10 20 25

```

var
  c1, c2, s: Inteiro
inicio
  s <- 0
  c1 <- 0
  Enquanto (c1 < 3) faça
    Para c2 <- 1 ate 9 passo 3 faça
      s <- s + 1
    FimPara
    c1 <- c1 + 1
  FimEnquanto
  Escreva (s)
finalgoritmo

```

s	c1	c2
0	0	1
1	1	4
2	2	7
3	3	10
4	4	13
5	5	16
6	6	19
7	7	22
8	8	25
9	9	28
10	10	31
11	11	34
12	12	37
13	13	40
14	14	43
15	15	46
16	16	49
17	17	52
18	18	55
19	19	58
20	20	61
21	21	64
22	22	67
23	23	70
24	24	73
25	25	76
26	26	79
27	27	82
28	28	85
29	29	88
30	30	91
31	31	94
32	32	97
33	33	100
34	34	103
35	35	106
36	36	109
37	37	112
38	38	115
39	39	118
40	40	121
41	41	124
42	42	127
43	43	130
44	44	133
45	45	136
46	46	139
47	47	142
48	48	145
49	49	148
50	50	151
51	51	154
52	52	157
53	53	160
54	54	163
55	55	166
56	56	169
57	57	172
58	58	175
59	59	178
60	60	181
61	61	184
62	62	187
63	63	190
64	64	193
65	65	196
66	66	199
67	67	202
68	68	205
69	69	208
70	70	211
71	71	214
72	72	217
73	73	220
74	74	223
75	75	226
76	76	229
77	77	232
78	78	235
79	79	238
80	80	241
81	81	244
82	82	247
83	83	250
84	84	253
85	85	256
86	86	259
87	87	262
88	88	265
89	89	268
90	90	271
91	91	274
92	92	277
93	93	280
94	94	283
95	95	286
96	96	289
97	97	292
98	98	295
99	99	298
100	100	301



```

var
  m, n, z: Inteiro
inicio
  m <- 300
  n <- 50
  z <- 0
  Enquanto (z < 200) faca
    Se (m < 100) entao
      z <- z + 100
    senao
      Se (m < 200) entao
        z <- z + 10
      senao
        z <- z + n
      FimSe
    FimSe
    → m <- m - n
  FimEnquanto
  Escreva (m, n, z)
finalgoritmo

```

M	N	Z
300	50	0
250		50
200		100
150		150
100		160
50		170
0		270

m n z
 0 50 270

```

var
  peso: Real
  fator: Inteiro
  acabou: Logico
inicio
  peso <- 3.0
  fator <- 4
  acabou <- falso
  Enquanto (nao acabou) faca
    Se (peso >= 10) entao
      peso <- peso  $\Delta$  fator
    FimSe
    peso <- peso + fator
    fator <- fator - 1
    Se (peso > 100) entao
      acabou <- verdadeiro
    FimSe
  FimEnquanto
  Escreva(peso)
finalgoritmo

```

10.0^2
 1000

Peso	Fator	Acabou
3.0	4	Falso
7.0	3	VERDADEIRO
10.0	2	
100.0	1	
102.0	0	

EX PONENCIAÇÃO
 PESO $\uparrow$ FATOR

102.0

```

var
  a, b, c, d: Inteiro
inicio

```

```

  a <- 3

```

```

  b <- 1

```

```

  c <- a - b

```

```

  d <- a % c

```

```

  Enquanto (d <= 4) faça

```

```

    Se ((a >= 3) e nao (d < b) xou (c > 4)) entao

```

```

      a <- a + 1

```

```

      b <- a

```

```

    senao

```

```

      c <- c + 3

```

```

    FimSe

```

```

    d <- d + 1

```

```

  FimEnquanto

```

```

  Escreva (a, b, c, d)

```

```

finalgoritmo

```

A	b	c	D
3	1	2	2
4	2	5	3
5	3		4
6	6		5

16 % 3 = 1

16 / 3 = 5
 16 - 15 = 1

A	b	c	D
6	6	5	5

```

var
  a, b: Inteiro
inicio
  a <- 10
  b <- 5
  Enquanto (a < 20000) faca
    Se (a > 30) entao
      interrompa →
    senao
      Se (a > 20) entao
        a <- a + 2
      senao
        a <- a + b
      FimSe
    FimSe
  FimEnquanto
  Escreva (a)
fimalgoritmo

```

A.	B
10	5
15	
20	
25	
27	
29	
<u>31</u>	

(31)

num

0	1	1	3	2	1	0
1	2	3	4	5	6	7

i	c
7	1
6	1
5	2
4	3
3	1
2	1
1	0

```

var
  num: vetor[1..7] de Inteiro
  i, c: Inteiro
inicio
  i <- 7
  Para c <- 1 ate 7 faca
    num[c] <- i % c
  FimPara
  Para i <- 1 ate 7 faca
    Escreva(num[i])
  FimPara
finalgoritmo
  
```

ALIMENTANDO

ESCREVENDO

0 1 1 3 2 1 0

```

var
  N: Vetor[1..4] de Inteiro
  C: Inteiro
inicio
  Para C <- 1 ate 4 faca
    N[C] <- C * 3
  FimPara
  Para C <- 4 ate 2 passo -1 faca
    N[C] <- N[C-1] + 1
  FimPara
  Para C <- 1 ate 4 faca
    Escreva(N[C])
  FimPara
fimalgoritmo

```

N

	4	7	10
1	3	6	9
2			12
3			
4			

2
3
4
5
6
7
8
9
10
11
12

3 4 7 10

F

0	1	1	2	3	5	8	13	21	34
1	2	3	4	5	6	7	8	9	10

```

var
    f: vetor[1..10] de Inteiro
    i, c: Inteiro
inicio
    f[1] <- 0
    f[2] <- f[1] + 1
    f[3] <- f[2]
    Para c <- 4 ate 10 faca
        f[c] <- f[c-1] + f[c-2]
    FimPara
    Para c <- 1 ate 10 faca
        Escreva(f[c])
    FimPara
fimalgoritmo
  
```

0
1
1
2
3
5
8
13
21
34

0 1 1 2 3 5 8 13 21 34

Exemplo **procedimento com parâmetros**
algoritmo "procedimento com parâmetros"

```
var n, m, res: inteiro
  procedimento soma (x,y: inteiro)
  inicio
    // n, m e res são variáveis globais
    res <- x + y
  fimprocedimento
inicio
// Seção de Comandos
  n <- 4
  m <- 9
  soma (n,m)
  escreva(res)
finalgoritmo
```

```
algoritmo "Exemplo de passagem de parâmetro por referência"
var
  a, b:inteiro
  procedimento troca (var x: inteiro; var y:inteiro)
  var aux:inteiro
  inicio
    aux <- x
    x <- y
    y <- aux
  fimprocedimento
inicio
  escreva ("entre com a: ")
  leia(a)
  escreva ("entre com b: ")
  leia(b)
  troca(a,b)
  escreval ("Valor de a: ",a)
  escreval ("Valor de b: ",b)
finalgoritmo
```

