



LÓGICA DE PROGRAMAÇÃO AULA 4

+55 (21) 99461-8818

@explicadoresnet

www.explicadores.net.br



- Assinale a alternativa que contém o valor final da variável X, após a execução do trecho de programa em Português Estruturado mostrado abaixo. Considere os seguintes valores iniciais para as variáveis: $X = 1$; $A = 5$.

enquanto ($A < 9$) faça

se ($X < 6$) então

$X \leftarrow X * 2$

fim_se

$A \leftarrow A + 1$

fim_enquanto

- a) 4
- b) 6
- ☒ c) 8
- d) 16

$A \geq 9$

condição

FLAG

condição de
término do
programa
ou
estrutura

ENQUANTO (V) faça Repita
até (V)
fim enquanto

Para

X
1
2
4
8
16

A
5
6
7
8
9

84 – Associe a coluna da direita com a da esquerda e, em seguida, marque a alternativa que contém a seqüência correta.

- | | |
|--|---|
| 1- Desvio Condicional Simples | (3) enquanto ... faça ...
fim_enquanto |
| 2- Desvio Condicional Composto | (1) se ... então ... fim_se |
| 3- Repetição do Tipo: Teste lógico no <u>início</u> do looping | (4) <u>repita</u> ... até_que |
| 4- Repetição do Tipo: Teste lógico no <u>fim</u> do looping | (5) { para ... de ... até ...
passo ... faça ...
fim_para |
| 5- Repetição do <u>Tipo</u> : <u>Variável</u> de <u>controle</u> | (2) <u>se</u> ... então ... senão ... |

- a) 1, 2, 3, 5, 4
- ~~b)~~ 3, 1, 4, 5, 2
- c) 3, 5, 4, 1, 2
- d) 4, 2, 3, 5, 1

– Assinale a alternativa que contém o texto impresso no final da execução do programa em Português Estruturado abaixo. Considere o valor inicial da variável X = 5.

→ X ← X + 2
 se (X = 5) então
 escreve "sete" *simples*
 fim_se
 se (X = 7) então
 escreve "cinco" *simples*
 fim_se
 Y ← 0 *2 7*
 enquanto (Y < X) faça
 escreve "V"
 Y ← Y + 1
 fim_enquanto

b

~~a) cincoVVVVVVVV~~
~~b) seteVVVVVVVV~~
 c) cincoVVVVVV
~~d) seteVVVVVV~~

X
 5
 7

Y
 0
 1
 2
 3
 4
 5
 6
 7

cincoVVVVVVVV
VVVVVVVVV

- Interprete o algoritmo e responda qual seqüência de números será impressa:

```
programa prova
var
  x,r : inteiro
  cont: inteiro
inicio
  cont ← 1
  x ← 0
  enquanto (cont <= 5) faça
    r ← x*3
    escreva r
    cont ← cont + 1
    x ← x+1
  fim-enquanto
fim
```

- a) 3,6,9,12,15
- b) 0,9,12,15,30
- c) 0,6,9,12,15
- ~~d) 0,3,6,9,12~~

Handwritten notes:

CONT	X	R
1	0	0
2	1	3
3	2	6
4	3	9
5	4	12

0, 3, 6, 9, 12

– Analise o algoritmo do programa abaixo, em português estruturado, e assinale qual será o valor final da variável X no momento da escrita.

programa SURPRESA

var

CONTADOR : inteiro

X : inteiro

inicio

X $\leftarrow$ 1

CONTADOR $\leftarrow$ 1

enquanto (CONTADOR $\leq$ 5) faça

X $\leftarrow$ X * CONTADOR

CONTADOR $\leftarrow$ CONTADOR + 1

fim_enquanto

escreva "O valor de X é = ", X SAÍDA

fim

a) 5

b) 24

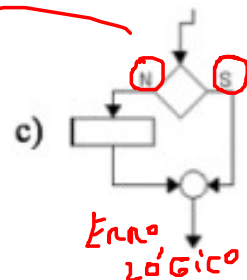
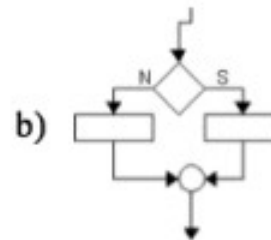
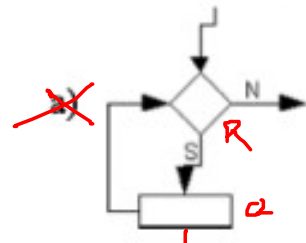
c) 1

~~d) 120~~

X	CONTADOR
1	1
2	2
4	3
8	4
24	5
120	6

– Assinale a alternativa que apresenta o diagrama de blocos descrito a seguir.

“Caracteriza-se por uma estrutura que efetua um teste lógico no início de um looping, verificando se é permitido executar o trecho de instruções subordinado a esse looping”.



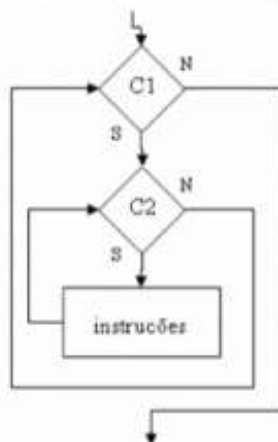
ENQUANTO

CONDICIONAL COMPOSTO

NENHUMA

Teste lógico

– Selecione a alternativa com o trecho de programa em português estruturado correspondente ao diagrama de blocos da figura abaixo.
 Considere: C1 = Condição 1 e C2 = Condição 2.



~~a) enquanto (<C1>) faça
 enquanto (<C2>) faça
 <instruções>
 fim_enquanto
 fim_enquanto~~

b) enquanto (<C1>) faça
 repita
 <instruções>
 até_que (<C2>)
 fim_enquanto

c) repita
 repita
 <instruções>
 até_que (<C2>)
 até_que (<C1>)

d) enquanto (<C1> e <C2>) faça
 para <var> de <início> até <fim> passo <incr> faça
 <instruções>
 fim_para
 fim_enquanto

– Assinale a alternativa que apresenta o que o programa SOMATÓRIO retornará na variável “TOTAL”:

Programa SOMATÓRIO

Var

TOTAL, CONTADOR: inteiro

Início

TOTAL $\leftarrow$ 0

para CONTADOR de 1 até 50 passo 1 faça

TOTAL $\leftarrow$ TOTAL + CONTADOR

Fim-para

Escreva TOTAL

Fim

- a) Será retornado o ~~valor~~ 50.
- b) Será retornada a média aritmética dos primeiros 50 números inteiros.
- c) Não há como prever qual valor será retornado, pelo valor tender ao infinito.
- ~~d)~~ Será retornada a soma dos 50 primeiros números inteiros (1+2+3+4+ ... +47+48+49+50).

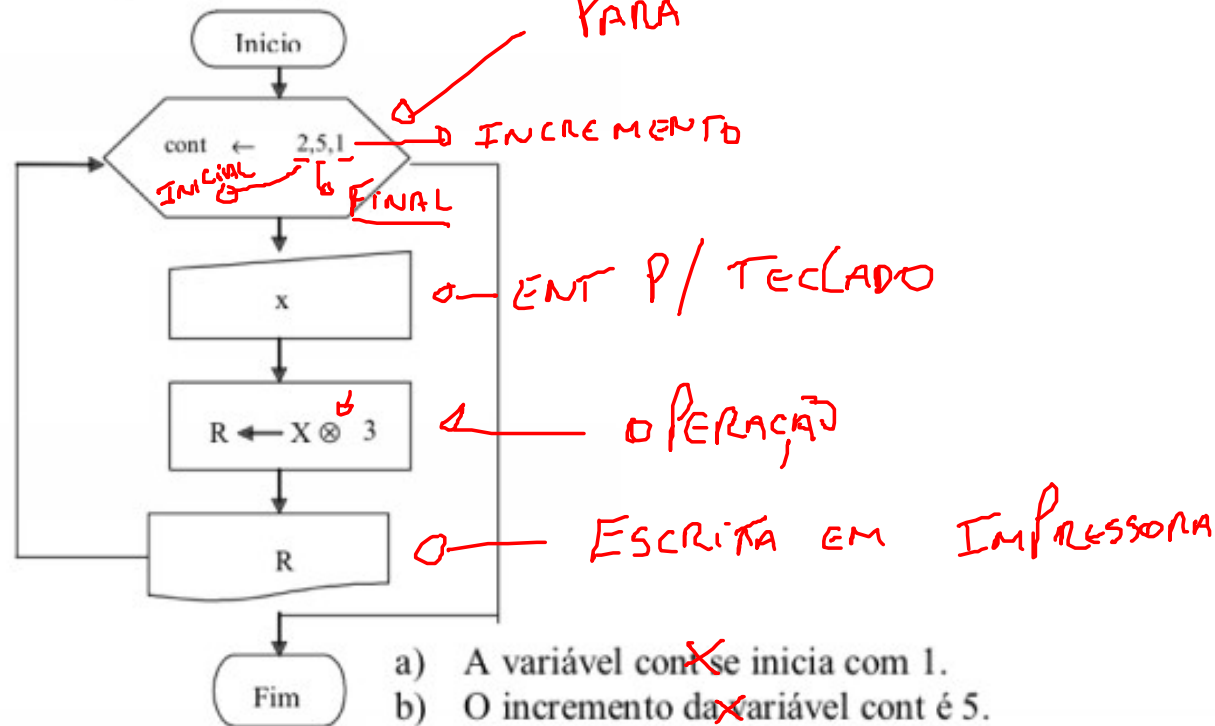
CONTADOR

1
3
5
7
9
11
...

1
2
3
4
5
6
7
8
...

TOTAL	CONTADOR
0+1	1
0+2	2
0+3	3
...	...
...	50

– Assinale a alternativa que retrata corretamente a utilização da estrutura de *looping* na figura abaixo, estrutura para...de...até...passo...faça...fim...para:



- a) A variável ~~cont~~ se inicia com 1.
- b) O incremento da ~~variável~~ cont é 5.
- c) A estrutura de repetição ~~será~~ executada 5 vezes.
- ~~d)~~ A estrutura de repetição pode ser usada quando houver necessidade de se repetir trechos finitos, em que os valores “inicial” e “final” são conhecidos.

```

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

```

15
 19
 30 17 315

S	C
0	1
4	5
8	9
12	13

15

```

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

```

X	Y	Z
1	0	8
2	8	5
3	13	8
4	21	5
5		
		26

```

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

```

c	i	x
0	5	0
10	↑	5
20		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 ((5))
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

```

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 ^ fator
    FimSe
    peso <- peso + fator
    fator <- fator - 1
    Se (peso > 100) entao
      acabou <- verdadeiro
    FimSe
  FimEnquanto
  Escreva(peso)
finalgoritmo

```

Peso	fator	acabou
3.0	4	X
7.0	3	-
10.0	2	✓
100.0	1	
102.0		

simples

F

10.0²

102.0

3 2
1

```
var
  a, b, c, d: Inteiro
```

```
inicio
```

```
  a <- 3
```

```
  b <- 1
```

```
  c <- a - b
```

```
  d <- a % c
```

```
  Enquanto (d <= 4) faca
```

```
    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)
```

```
fimalgoritmo
```

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

✓

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)
finalgoritmo

```

Handwritten notes:

- A red box around "interrompa" with an arrow pointing to it and the text "Zerou a 0" and "LACO" written next to it.
- A red circle around the number "31" at the end of the program.
- Red curved lines indicating the flow of the loop and the final output.

A	B
10	5
15	
20	
25	
27	
29	
31	