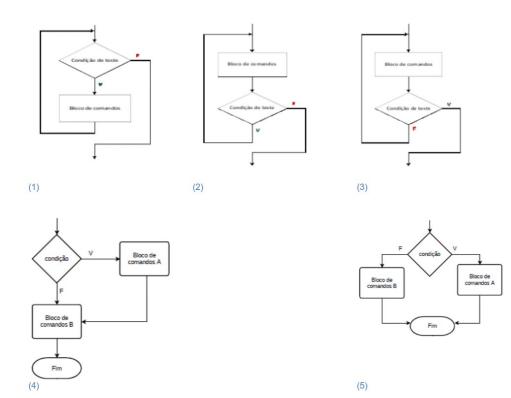
| Usuário | | |
|---------------------------|------------------------------------|--|
| Curso | LÓGICA DE PROGRAMAÇÃO E ALGORITMOS | |
| Teste | QUESTIONÁRIO UNIDADE II | |
| Iniciado | 03/11/22 07:59 | |
| Enviado | 03/11/22 08:12 | |
| Status | Completada | |
| Resultado da tentativa | 2,5 em 2,5 pontos | |
| Tempo decorrido | | |
| Resultados exibidos | ,,,, | |
| D | - A | |

Pergunta 1

0,25 em 0,25 pontos

Analise os fluxogramas dados abaixo e assinale a alternativa que representa corretamente o propósito:



Resposta Selecionada: **⊘**a.

Laço ENQUANTO, Laço FACA ... ENQUANTO, Laço REPITA ... ATE, decisão simples SE... ENTAO, decisão composta SE.. SENAO.

Respostas:

⊘a.

Laço ENQUANTO, Laço FACA ... ENQUANTO, Laço REPITA ... ATE, decisão simples SE... ENTAO, decisão composta SE.. SENAO.

b.

Laço FACA ... ENQUANTO, Laço ENQUANTO, Laço REPITA ... ATE, decisão simples SE... ENTAO, decisão composta SE.. SENAO.

C.

Laço ENQUANTO, Laço REPITA ... ATE, Laço FACA ... ENQUANTO, decisão simples SE... ENTAO, decisão composta SE.. SENAO.

d.

Decisão simples SE... ENTAO, decisão composta SE.. SENAO, Laço ENQUANTO, Laço FACA ... ENQUANTO, Laço REPITA ... ATE.

e.

Laço REPITA ... ATE, Laço ENQUANTO, Laço FACA ... ENQUANTO, decisão simples SE... ENTAO, decisão composta SE.. SENAO.

Comentário da resposta:

Resposta: A

Comentário: O fluxograma (1) é de um laço de repetição com teste no início, sendo executado o bloco de código enquanto a condição for verdadeira.

O fluxograma (2) é de um laço de repetição com teste no fim. O bloco é executado pelo menos uma vez, visto que apenas após a sua execução, a condição será testada. O bloco será repetido enquanto a condição for verdadeira. O fluxograma (3) é de um laço de repetição com teste no fim, visto que o bloco será executado antes da condição ser testada. O bloco será repetido até que a condição seja verdadeira. É uma interpretação sutilmente diferente do fluxograma (2), visto que nesta lógica, o resultado falso para a condição força a permanência no laço.

O fluxograma (4) é de uma estrutura de decisão simples. Se a condição for verdadeira, então o bloco A será executado, mas se for falsa, o algoritmo seguirá seu fluxo normal. Ou seja, haverá um desvio apenas se a condição for verdadeira.

O fluxograma (5) é de uma estrutura de decisão composta. Se a condição for verdadeira, então o bloco A será executado, senão o Bloco B será executado, nunca ambos.

· Pergunta 2

0,25 em 0,25 pontos



Analise o algoritmo dado abaixo e responda quanto vale x ao término da execução.

```
Algoritmo "Unidade 2 - Questao2"
2.
3.
4.
5.
6.
7.
8.
9.
             Var
                    i, j, x : caractere
             Início
                    //entrada
                    x < 0
                    para i de 10 ate 1 passo 1 faça
                          para j de 1 ate 10 passo 1 faca
                               \underline{se} (i mod 2 = 0) \underline{então}
                                      x < -x+1
11.
                               senão
12.
                                       x < -x-1
13.
                               fimse
14.
                          fimpara
15.
                    fimpara
16.
                    escreva(x)
17.
             Fimalgoritmo
```

Resposta Selecionada: 🛂a.

0.

Respostas:



0.

b. 2.

c.

10.

d.

50.

e.

100.

Comentário da resposta:

Resposta: A

Comentário: A melhor forma de compreender a saída de um código é fazendo o teste de mesa. Na primeira iteração do laço para da linha 7, a variável i vale 10, que é par, logo a variável x será incrementada 10 vezes no laço interno controlado pela variável j. Na segunda iteração do laço externo, a variável i valerá 9, que é ímpar, e será decrementada 10 vezes. A cada 10 incrementos, são feitos 10 decrementos. Como a variável i está num laço de 10 até 1, será executado 10 vezes, portanto, a única chance da variável x guardar um valor diferente de zero, é ser o laço externo fosse executado um número ímpar de vezes, o que não é o caso.

· Pergunta 3

0,25 em 0,25 pontos



O algoritmo "Dias de Aula" é projetado para ler um número que representa determinado dia da semana e, após a leitura, escreve o dia por extenso e a disciplina do dia, ou a mensagem "Dia Inválido", caso o número digitado não esteja na faixa de 1 a 7.

PORQUE

A estrutura ESCOLHA... CASO oferece uma forma organizada para agrupar os comandos e a lógica consiste em verificar o valor da variável que controlará a decisão, e uma ação diferente será executada para cada valor que a variável poderá assumir.

```
Algoritmo "Dias de Aula"
2.
      Var
3.
         dia da semana : inteiro
4.
      Inicio
        escreva ("*=====
5.
         escreva ("*
                        DIA DA SEMANA
         escreva ("* [1] Domingo
7.
        escreva("* [2] Segunda-feira
8.
         escreva("* [3] Terça-feira
9.
         escreva("* [4] Quarta-feira
10.
11.
                                                 *")
         escreva("* [5] Quinta-feira
         escreva ("* [6] Sexta-feira
                                                 * 11
12.
                                                 *")
         escreva ("* [7] Sábado
13.
         escreva ("*=
14.
15.
         leia (dia_da_semana)
        escolha (dia da semana)
16.
17.
             caso 1
                  escreva ("Domingo não tem aula")
18.
19.
20.
                   escreva ("Aulas da Segunda-feira")
21.
                   escreva ("Lógica de Programação e Algoritmos")
22.
             caso 3
23.
                   escreva ("Aulas da Terça-feira")
                   escreva ("Lógica Matemática")
24.
25.
             caso 4
                   escreva ("Aulas da Quarta-feira")
26.
                   escreva ("Interação Humano-Computador")
27.
28.
29.
                   escreva ("Aulas da Quinta-feira")
30.
                   escreva ("Introdução à Programação Estruturada")
31.
            caso 6
                   escreva ("Aulas da Sexta-feira")
32.
33.
                   escreva ("Progr. Dispositivos Móveis")
             caso 7
34.
                   escreva ("Aulas do Sábado")
35.
                   escreva ("autoestudo")
36.
37.
             outrocaso
38.
                   escreva ("Dia inválido!")
39.
         fimescolha
40.
      Fimalgoritmo
```

Assinale a alternativa correta:

Resposta 🛂a

Selecionada: As asserções I e II são verdadeiras, e a asserção II justifica a asserção I.

Respostas: Va

As asserções I e II são verdadeiras, e a asserção II justifica a asserção I.

b.

As asserções I e II são verdadeiras, e a asserção II não justifica a asserção I.

c.

A asserção I é verdadeira e a asserção II é falsa.

d.

A asserção I é falsa e a asserção II é verdadeira.

e.

As asserções I e II são falsas.

Comentário: o algoritmo implementa uma estrutura de decisão que seleciona o algoritmo de acordo com o valor da variável "dia_da_semana". Como há 7 dias na semana, o algoritmo apresenta um bloco de comandos contendo o dia da semana por extenso e a disciplina do dia. Se o usuário digitar um número que não corresponde a um dia da semana válido, o algoritmo está projetado para responder que o dia é inválido.

Comentário da resposta: Resposta: A

· Pergunta 4

0,25 em 0,25 pontos



Quanto vale k no final do seguinte procedimento:

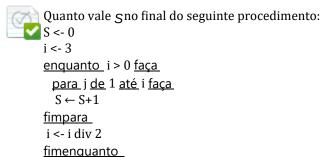
```
k <- 0
para i de 1 ate
n faca
   <u>pa</u>ra j de <u>1</u> ate <u>n</u> faca
     k <- k+1
     <u>fimpar</u>a
fimpara
Resposta Selecionada: Q_{C}
                       n 2.
Respostas:
                       10.
                       b.
                       n.
                       ⊘c.
                       n².
                       d.
                       n³.
                       e.
                       n/2.
```

Comentário da resposta:

Resposta: C

Comentário: Para cada iteração do laço externo, que varia de 1 a n, haverá n iterações no laço interno, que também varia de 1 até n.

O algoritmo executará n x n vezes, ou seja, a



Resposta Selecionada: 💁.

4.

Respostas:

a. 2.

b.

3.

4.

d.

5.

e.

6.

Comentário Resposta: C

da

resposta:

Comentário: A tabela abaixo é o teste de mesa para o algoritmo. Na primeira iteração do laço enquanto, a variável i vale 3 e o laço para é repetido três vezes, e ao término do laço para, a variável S vale 3. Ao finalizar o laço para, a variável i recebe o quociente da divisão i/2, cujo resultado é 1. Com i valendo 1, a condição do laço enquanto é satisfeita e o bloco interno é executado. O laço para vai variar a variável j de 1 até i, logo de 1 até 1, executando uma única vez. Dentro deste laço, a variável S será incrementada e valerá 4 e a variável i receberá o quociente da divisão i/2, como i vale 1, o quociente da divisão é 0 e a condição do laço enquanto resulta Falso, interrompendo a execução do algoritmo.

Isto posto, ao término, S vale 4.

| i>0 | i | j | S |
|-----|-------------------------------|---|---|
| V | 3 | | 0 |
| | 77 | 1 | 1 |
| | | 2 | 2 |
| | | 3 | 3 |
| V | i ←i div 2 i ←1 div 2 1 | 1 | 4 |
| F | i ←i div 2 i ←1 div 2 0 | | |

· Pergunta 6

0,25 em 0,25 pontos



```
Algoritmo "Incognita 01"
                                                    Algoritmo "Incognita 02"
Var
                                                    Var
 x, num: inteiro
                                                     num: inteiro
Inicio
                                                    Inicio
                                                      escreva("Numero = ")
  escreva("x = ")
                                                       leia(x)
   leia(x)
  num <- x mod 10
  se (num=0 ou num=2 ou num=4 ou num=6
                                                      \underline{se} (x mod 2 = 0) então
ou num=8)
                                                        escreva(x, "eh par")
    escreva(num, "eh par")
                                                      senão
                                                        escreva(x, "eh impar")
  senão
    escreva(num, "eh impar")
                                                      fimse
                                                    Fimalgoritmo
Fimalgoritmo
```

- I. O algoritmo Incognita 01 compara o último dígito do número armazenado em x e, se for 0, 2, 4, 6 ou 8, então escreve a mensagem x é par, senão escreve a mensagem x é ímpar.
- II. Os algoritmos Incognita 01 e 02 produzem saídas iguais, independentemente do valor de x. III. O algoritmo Incognita 02 verifica se o quociente da divisão de x por 2 é 0 e, em caso afirmativo escreve a mensagem x é par, senão escreve x é ímpar.
- IV. Os algoritmos Incognita 01 e 02 produzem saídas completamente diferentes quando o valor armazenado na variável x é par ou ímpar.

Assinale a alternativa correta.

Resposta Selecionada: \bigcirc_{a}

As afirmativas I e II são verdadeiras.

Respostas:

⊘a.

As afirmativas I e II são verdadeiras.

As afirmativas II e III são verdadeiras.

Apenas a afirmativa II é verdadeira.

Apenas a afirmativa III é verdadeira.

Apenas a afirmativa IV é verdadeira.

Comentário Resposta: A da resposta: Comentário:

I. Afirmativa correta.

O algoritmo Incognita 01 lê um número inteiro e armazena na variável x. O comando num <- x mod 10 armazenará em num o resto da divisão de x por 10. Quando se divide um número por 10, o resto da divisão é o último dígito do número e, ao pegar o último dígito do número e compará-lo com 0, 2, 4, 6 ou 8, o algoritmo escreve num é par em caso verdadeiro, ou num é ímpar em caso falso.

II. Afirmativa correta.

Considerando a justificativa da afirmativa I e, o algoritmo Incognita 02 lê um número inteiro e armazena na variável x. O comando x mod 2 armazenará em num o resto da divisão de num por 2. Se a divisão for exata o resultado será 0 e todo número divisível por 2 é par.

Logo, os algoritmos Incognita 01 e 02 produzem saídas iguais,

independentemente do valor x.

III. Afirmativa incorreta.

A condição do comando se, (x mod 2 = 0), testa se o resto da divisão de x por 2 é igual a zero, e não o quociente da divisão conforme declarado. IV. Afirmativa incorreta.

Independentemente da entrada para a variável x, ambos os algoritmos apresentam as mesmas saídas.

· Pergunta 7

0,25 em 0,25 pontos



Pela regra clássica da divisibilidade, um número é divisível por 3 se a soma dos seus 🔽 algarismos é divisível por 3.

Exemplos:

```
27 é divisível por 3 pois 2+7=9 que é divisível por 3.
576 é divisível por 3 pois 5+7+6=18 e 18=1+8=9 que é divisível por 3.
134 não é divisível por 3, pois 1+3+4=8 que não é divisível por 3.
```

Analise os algoritmos abaixo e selecione o que implementa corretamente a regra da

```
divisibilidade por 3.
Resposta Selecionada: 🛂a.
                        Algoritmo "Divisibilidade por 3"
                           x, soma, num: inteiro
                         Inicio
                         soma <- 0
                         escreva ("Numero: ")
                         <u>leia (</u>num)
                         enquanto (num > 0) faca
                            x <- num <u>mod</u> 10
                            soma <- soma + x
                            num <- num div 10
                            \underline{se} ((num < 1) \underline{e} (soma > 10)) entao
                                num <- soma
                               soma <- 0
                          fimse
                        <u>fimenquanto</u>
                        se ((soma = 3) ou (soma = 6) ou (soma = 9)) então
                         escreva (num, "eh divisível por 3")
                         escreva (num, "NAO eh divisível por 3")
                        <u>fimse</u>
                        <u>Fimalgoritmo</u>
Respostas:
                        ⊘a.
                        Algoritmo "Divisibilidade por 3"
                           x, soma, num: inteiro
                         <u>Inicio</u>
                         soma <- 0
                         escreva ("Numero: ")
                         leia (num)
                         enquanto (num > 0) faca
```

x <- num <u>mod</u> 10

```
soma <- soma + x
    num <- num div 10
    <u>se</u> ((num < 1) <u>e</u> (soma > 10)) entao
        num <- soma
        soma <- 0
  <u>fimse</u>
fimenquanto
se ((soma = 3) ou (soma = 6) ou (soma = 9)) então
 escreva (num, "eh divisível por 3")
senão
 escreva (num, "NAO eh divisível por 3")
<u>fimse</u>
<u>Fimalgoritmo</u>
b.
Algoritmo "Divisibilidade por 3"
Var
x, soma, num : inteiro
<u>Inicio</u>
soma <- 0
escreva ("Numero: ")
leia (num)
enquanto (num > 0) faca
x <- num <u>mod</u> 10
soma <- soma + x
num <- num div 10
fimenquanto
<u>se</u> ((soma = 3) <u>ou</u>
(soma = 6) <u>ou</u> (soma = 9)) <u>então</u>
 escreva (num, "eh divisível por 3")
senão
 escreva (num, "NAO eh divisível por 3")
<u>fimse</u>
<u>Fimalgoritmo</u>
c.
Algoritmo "Divisibilidade por 3"
<u>Var</u>
x, soma, num: inteiro
<u>Inicio</u>
soma <- 0
escreva ("Numero: ")
leia (num)
enquanto (num > 0) faca
x <- num <u>mod</u> 10
soma <- soma + x
num <- num div 10
\underline{\mathsf{se}}(\mathsf{num} < 1) \ \mathsf{e} \ (\mathsf{soma} > 10)) \ \mathsf{entao}
num <- soma
soma <- 0
<u>fimse</u>
<u>fimenquanto</u>
```

```
\underline{se} ((soma = 3) \underline{e}
(soma = 6) <u>e</u> (soma = 9)) <u>então</u>
 escreva (num, "eh divisível por 3")
<u>senão</u>
 escreva (num, "NAO eh divisível por 3")
<u>fimse</u>
<u>Fimalgoritmo</u>
d.
Algoritmo "Divisibilidade por 3"
<u>Var</u>
i, x, soma, num : inteiro
Inicio
soma <- 0
escreva ("Numero: ")
leia (num)
num <- num/3
enquanto (num > 0) faca
x <- num <u>mod</u> 10
soma <- soma + x
num \leftarrow num div 10
<u>se</u> ((num < 1) <u>e</u>
(soma > 10)) entao
num <- soma
soma <- 0
<u>fimse</u>
fimenquanto
<u>se</u> ((soma = 3) <u>ou</u>
(soma = 6) ou (soma = 9)) então
 escreva (num, "eh divisível por 3")
senão
 escreva (num, "NAO eh divisível por 3")
<u>fimse</u>
<u>Fimalgoritmo</u>
Algoritmo "Divisibilidade por 3"
<u>Var</u>
i, x, soma, num : inteiro
<u>Inicio</u>
soma <- 0
escreva ("Numero: ")
leia (num)
num <- num/2
enquanto (num > 0) faca
x <- num mod 10
soma <- soma + x
num <- num div 10
<u>se</u> ((num < 1) <u>e</u>
(soma > 10)) entao
num <- soma
soma <- 0
```

```
fimse
fimenquanto
se ((soma = 3) ou
(soma = 6) ou (soma = 9)) então
escreva (num, "eh divisível por 3")
senão
escreva (num, "NAO eh divisível por 3")
fimse
Fimalgoritmo
```

Comentário da resposta:

Resposta: A

Comentário: O algoritmo da alternativa A está correto porque lê um número inteiro e, enquanto este número for maior que zero, pega o último dígito, soma-o e retira o último dígito do número. Repete este processo enquanto o número é maior que 0. Se o número for menor que 1 e a soma for maior que 10 então o número recebe a soma e a soma recebe 0. Ao término deste processo, se o número resultante terminar em 3, 6 ou 9 o número é divisível por 3.

O algoritmo da alternativa B está incorreto porque faltou o trecho de código abaixo, o qual é responsável por redefinir os valores de num e soma para os novos valores, necessários para que o algoritmo seja repetido até que o valor seja um número de 1 dígito.

```
se ((num < 1) e (soma > 10)) então
num <- soma
soma <- 0
fimse</pre>
```

A alternativa C está errada porque a estrutura de decisão nunca será verdadeira com o operador E-Lógico. A soma será 3 ou 6 ou 9, nunca ambos.

```
\underline{se} ((soma = 3) \underline{e} (soma = 6) \underline{e} (soma = 9))
```

A alternativa D está errada porque ao dividir o número por 3, o algoritmo funcionará apenas para os números divisíveis por 9, devido à inclusão do comando num <- num/3 após o comando <u>leia (num)</u>.

Lembrando que um algoritmo é considerado correto quando apresenta resultados corretos para todas as instâncias do problema, a alternativa e está errada porque ao dividir o número por dois no comando num <- num/2 após o comando <u>leia (num)</u>, o número verificado será modificado e o algoritmo não funcionará para todas as instâncias de valores que efetivamente são divisíveis por 3.

· Pergunta 8

0,25 em 0,25 pontos



Analise o algoritmo dado abaixo e considere que as variáveis lógicas assumem os valores.

VERDADEIRO ou FALSO.

Algoritmo "Unidade II – Questão 10" Var A, B, C, D, E, F, G: lógico num, x: inteiro Inicio leia (A, B, C, D, E, F, G)

se ((A ou B) e (C ou D)) então

```
num <- 1
x <- 10
<u>senão</u>
num <- 2
x <- 14
fimse
escolha (num)
caso 1
 faca
 escreva("7")
 num <- num-14
 enquanto (num>0)
<u>outrocaso</u>
 <u>enquanto</u>
(num>0) faca
 escreva("9")
 num <- num-1
 <u>fimenguanto</u>
<u>Fimalgoritmo</u>
```

Considere que o algoritmo foi executado duas vezes e, na primeira execução do algoritmo, foram atribuídas às variáveis lógicas os seguintes valores:

A ← VERDADEIRO

B ← FALSO

C ← VERDADEIRO

D ← FALSO

E na segunda execução, foram atribuídas às variáveis lógicas os seguintes valores:

(A) ← VERDADEIRO

(B) ← VERDADEIRO

(C) ← FALSO

(D) ← FALSO

A saída apresentada em cada execução foi, respectivamente

Resposta Sa

Selecionada: Primeira execução 7 e na segunda 99.

Primeira execução 7 e na segunda 99.

b.

Primeira execução 99 e na segunda 7.

c.

Primeira execução 7 e na segunda nada será executado porque no caso

de num valer 2 não há programação.

d.

Primeira execução 7 e na segunda 14.

e.

Nada será escrito porque a variável num não está sendo lida.

Comentário da resposta: Resposta: A

x < -14

Comentário: Na primeira execução, considerando as atribuições □ VERDADEIRO; B □ FALSO; C □ VERDADEIRO e D FALSO, a expressão lógica ((A ou B) e (C ou D)) será VERDADEIRO e as variáveis num = 1 e x = 10. O algoritmo salta para o comando escolha, que considera o primeiro caso, dado que num é igual a 1. No caso de num valer 1, será escrito 7. Apesar de estar numa estrutura de repetição, o comando será escrito uma única vez, pois num num-14 e, como num vale 10, o resultado da subtração será -4 e o laço de repetição está condicionado à num>0, portanto, o laço será interrompido. Na segunda execução, considerando as atribuições A WERDADEIRO; B VERDADEIRO; C FALSO e D FALSO, a expressão lógica ((A ou B) e (C ou D)) será FALSA e o bloco de comandos do SENÃO será executado, ou seja, num <- 2

O comando escolha não trata o caso de num ser igual 2 e o bloco de comandos descritos como outrocaso é executado. Neste bloco há um laço de repetição que é executado enquanto a variável num é maior que zero. Ao entrar neste bloco, a variável num será decrementada e este bloco será executado sempre duas vezes. O comando escreva será executado duas vezes, apresentando na tela o valor 99.

· Pergunta 9

0,25 em 0,25 pontos



Analise o algoritmo abaixo e escolha a alternativa que responde corretamente:

```
Algoritmo "Unidade 2 - Questao3"
2.
                  x, n: inteiro
4.
5.
6.
7.
            <u>Inicio</u>
                  //entrada
                  n < 0
                  para x de 20 ate 1 passo -2 faça
8.
                        x < -x - 1
9.
                        n <- n+1
10.
                        escreva(n)
11.
                  fimpara
12.
                   escreva(n)
13.
            Fimalgoritmo
```

- I. O laço de repetição terá executado 20 vezes.
- II. A seguinte sequência de números será escrita na tela: 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10.
- III. Ao término do algoritmo, a variável x vale 0.
- IV. De acordo com a lógica, a variável x assume apenas valores pares ou nulo.

Resposta Selecionada: Od.

Estão corretas as afirmativas III e IV.

Respostas:

Estão corretas as afirmativas I e II.

Estão corretas as afirmativas I e IV.

C

Estão corretas as afirmativas II e III.



Estão corretas as afirmativas III e IV.

e.

Apenas a IV está correta.

Comentário da resposta:

Resposta: D

Comentário:

I - Afirmativa incorreta.

Justificativa: O laço é iniciado com a variável x valendo 20 e irá ser executado até que x valha 1. No entanto, o passo -2 itera a variável x de 2 em 2 números. Na primeira iteração do laço, x vale 20. Na segunda, x vale 18. Na terceira iteração do laço, x vale 16, e nas próximas valerá 14, 12, 10, 8, 6, 4, 2 e 0. Quando x valer 0 o laço será interrompido, e terá sido executado 10 vezes.

II - Afirmativa incorreta.

Justificativa: Apesar da variável n ter sido inicializada com 0 na linha 6 do algoritmo, a primeira vez que o algoritmo escreve o seu valor é na linha 10 e, neste ponto, ela já terá sido incrementada na linha 9. Portanto, a sequência escrita pelo algoritmo não contém o número 0.

III - Afirmativa correta.

O laço é interrompido quando x vale 0, sendo este o último valor da variável x.

IV – Afirmativa correta.

Neste laço, a variável x será incrementada de 20 até 1, com passo -2, mas nesta lógica, esta variável nunca valerá 1.

· Pergunta 10

0,25 em 0,25 pontos

Deseja-se computar o número de votos a três candidatos à presidência da Associação Amigos do Bairro. Dado que há 50 moradores no bairro, o algoritmo abaixo implementa o processo de votação, mas falta o trecho de código que decide qual dos candidatos será eleito e o percentual de votos recebidos.

```
Algoritmo "Unidade 2 – Questao 4"
2.
                candidato, vencedor: caractere
4.
                votos_c1, votos_c2, votos_c3, votos_nulos: inteiro
                total moradores, total votos : inteiro
6.
7.
           <u>Inicio</u>
8.
                // inicialização das variáveis totalizadoras
9.
                total_moradores <- 50
10.
                total_votos <- 0
                votos_nulos <- 0
11
12.
                votos_c1 <- 0
                votos c2 <- 0
13.
                votos_c3 <- 0
14.
15.
16.
                repita
                      escreva("Para quem é o seu voto")
17
                      leia(candidato)
18.
                      escolha (candidato)
19
                      caso "C1"
20.
                          votos_c1 = votos_c1+1
21.
22
                      caso "C2"
23.
                         votos_c2 = votos_c2+1
24.
                      caso "C3"
25.
                          votos c3 = votos c3+1
26
                      outrocaso
27
                          votos nulos = votos nulos+1
28.
                      fimescolha
                      total_votos <- votos_c1 + votos_c2 + votos_c3 + votos_nulos
29
30.
                ate (total de votos < total_moradores)
31.
32
                // Quem venceu a eleição?
33.
34.
35.
           Fimalgoritmo
```

Escolha a alternativa que apresenta o trecho de código que completa o algoritmo. Assuma que o número de votos de cada candidato foi distinto.

```
Resposta
Selecionada:
               // quem é vencedor?
               se ((votos_c1>
               votos_c2) e (votos_c1>votos_c3) e (votos_c1>votos_nulos)) então
                      vencedor <- "C1"
               <u>senão</u>
                   se ((votos_c2>
               votos_c1) e (votos_c2>votos_c3) e (votos_c2>votos_nulos)) então
                      vencedor <- "C2"
                   <u>senão</u>
                      se ((votos_c3>
               votos_c1) e (votos_c3>votos_c2) e (votos_c3>votos_nulos) então
                            vencedor <- "C3"
                      <u>senão</u>
                            vencedor <- "NL"
                      <u>fimse</u>
                   <u>fimse</u>
               fimse
               se (vencedor="NL") então
                   escreval ("Eleição anulada!")
```

```
<u>senão</u>
                     escreval ("Candidato Vencedor : ", vencedor)
                 <u>fimse</u>
Respostas:
                 ⊘a.
                 // quem é vencedor?
                 se ((votos_c1>
                 votos_c2) <u>e</u> (votos_c1>votos_c3) <u>e</u> (votos_c1>votos_nulos)) <u>então</u>
                        vencedor <- "C1"
                 <u>senão</u>
                     se ((votos_c2>
                 votos_c1) e (votos_c2>votos_c3) e (votos_c2>votos_nulos)) então
                       vencedor <- "C2"
                     <u>senão</u>
                        se ((votos_c3>
                 votos_c1) e (votos_c3>votos_c2) e (votos_c3>votos_nulos) então
                              vencedor <- "C3"
                        <u>senão</u>
                              vencedor <- "NL"
                        <u>fimse</u>
                     <u>fimse</u>
                 <u>fimse</u>
                 se (vencedor="NL") então
                     escreval ("Eleição anulada!")
                     escreval ("Candidato Vencedor : ", vencedor)
                 fimse
                 // quem é vencedor?
                 se ((votos_c1>
                 votos_c2) <u>ou</u> (votos_c1>votos_c3) <u>ou</u> (votos_c1>votos_nulos)) <u>então</u>
                  vencedor <- "C1"
                 <u>senão</u>
                      se ((votos_c2>
                 votos_c1) <u>ou</u> (votos_c2>votos_c3) <u>ou</u> (votos_c2>votos_nulos)) <u>então</u>
                  vencedor <- "C2"
                      <u>senão</u>
                         se ((votos_c3>
                 votos_c1) <u>ou</u> (votos_c3>votos_c2) <u>ou</u> (votos_c3>votos_nulos) <u>então</u>
                         vencedor <- "C3"
                         <u>senão</u>
                         vencedor <- "NL"
                         <u>fimse</u>
                      fimse
                 fimse
                 se (vencedor="NL") então
                      escreval ("Eleição anulada!")
                 <u>senão</u>
```

```
escreval ("Candidato Vencedor: ", vencedor)
fimse
c.
// quem é vencedor?
se (votos_c3 < votos_c2 < votos_c1) então
vencedor <- "C1"
<u>senão</u>
     <u>se</u> (votos_c3 < votos_c1 < votos_c2) <u>então</u>
vencedor <- "C2"
     <u>senão</u>
       se (votos_c1 < votos_c2 < votos_c3) então</pre>
       vencedor <- "C3"
       <u>senão</u>
       vencedor <- "NL"
       <u>fimse</u>
     <u>fimse</u>
<u>fimse</u>
se (vencedor="NL") então
     escreval ("Eleição anulada!")
<u>senão</u>
     escreval ("Candidato Vencedor: ", vencedor)
fimse
// quem é vencedor?
se ((votos_c1>
votos_c2) e (votos_c1>votos_c3) e (votos_c1>votos_nulos)) então
vencedor <- "C1"
<u>senão</u>
     se ((votos_c2>
votos_c1) e (votos_c2>votos_c3) e (votos_c2>votos_nulos)) então
vencedor <- "C2"
     <u>senão</u>
       se ((votos_c3>
votos_c1) e (votos_c3>votos_c2) e (votos_c3>votos_nulos) então
       vencedor <- "C3"
       <u>fimse</u>
     <u>fimse</u>
<u>fimse</u>
se(vencedor="NL") então
     escreval ("Eleição anulada!")
<u>senão</u>
     escreval ("Candidato Vencedor: ", vencedor)
<u>fimse</u>
// quem é vencedor?
```

```
se ((votos_c1<
 votos_c2) e (votos_c1<votos_c3) e (votos_c1<votos_nulos)) então
  vencedor <- "C1"
 senão
       se ((votos_c2 <
 votos_c1) e (votos_c2<votos_c3) e (votos_c2<votos_nulos)) então
  vencedor <- "C2"
       <u>senão</u>
         se ((votos_c3<
 votos_c1) <u>e</u> (votos_c3<votos_c2) <u>e</u> (votos_c3<votos_nulos) <u>então</u>
         vencedor <- "C3"
         <u>senão</u>
         vencedor <- "NL"
         <u>fimse</u>
       <u>fimse</u>
 <u>fimse</u>
 se (vencedor <> "NL") então
       escreval ("Eleição anulada!")
 <u>senão</u>
       escreval ("Candidato Vencedor: ", vencedor)
 <u>fimse</u>
Resposta: A
```

Comentário da resposta:

Comentário: Neste exemplo de estrutura de decisão encadeada, para o algoritmo decidir quem é o mais votado, é necessário utilizar operadores lógicos e relacionais.

A variável A está sintática e logicamente correta.

A alternativa C está sintaticamente errada porque as expressões relacionais e lógicas são escritas na notação infixa, ou seja, <termo><operador><termo>.

A alternativa B está sintaticamente correta, mas errada porque nas expressões lógicas com disjunção, basta que uma expressão seja verdadeira para que a expressão seja verdadeira. Neste caso, mesmo que um dos candidatos seja mais votado em relação ao outro, mas não o mais votado de todos, atribuirá a variável vencedor o candidato avaliado, tornando o resultado errado.

A alternativa D está errada porque não avalia todas as situações possíveis.

A alternativa E está errada porque atribuiria à variável vencedor o candidato menos votado.