

LÓGICA DE PROGRAMAÇÃO E ALGORITMOS 7777-90_43701_R_E1_20222 CONTEÚDO

Revisar envio do teste: QUESTIONÁRIO UNIDADE IV

Usuário

Curso LÓGICA DE PROGRAMAÇÃO E ALGORITMOS

Teste QUESTIONÁRIO UNIDADE IV

Iniciado 22/11/22 21:12

Enviado 22/11/22 22:15

Status Completada

Resultado da tentativa 2,5 em 2,5 pontos

Tempo decorrido 1 hora, 2 minutos

Resultados exibidos Todas as respostas, Respostas enviadas, Respostas corretas, Comentários, Perguntas respondidas incorretamente

Pergunta 1

0,25 em 0,25 pontos



Analise o algoritmo dado abaixo e considere que usuário informou o valor 20 para o comando de entrada `leia(n)` apresentado na linha 11. Sabendo que o vetor `V` é indexado da seguinte forma

V

0	1	2	3

e, que cada linha das alternativas representa o valor que as variáveis assumem a cada iteração do algoritmo, assinale a alternativa que apresenta o valor correto que cada variável guarda a cada iteração.

```

1.      Algoritmo "Questao 1 Unidade 4"
2.      Var
3.      v[0..3], soma, i, j, n, :
4.      inteiro
5.      r : real
6.      Inicio
7.      i <- 0
8.      n <- 10
9.      soma <- 0
10.     para j de 3 até 0 passo -1 faça
11.     v[j] <- n
12.     n <- n + 10
13.     soma <- soma + v[j]
14.     i <- i+1
15.     fimpara
16.
17.     r <- soma/ i
18.     escreva("resultado = " + r)
19. Fimalgoritmo

```

Resposta
Selecionada:

✓ d.

i	j	n	soma
0		10	0
1	3	20	10
2	2	30	30
3	1	40	60
4	0	50	100

V

0	1	2	3

V

--	--	--	--

V

			10
--	--	--	----

V

		20	10
--	--	----	----

V

	30	20	10
--	----	----	----

V

40	30	20	10
----	----	----	----

r
25

Respostas:

a.

i	j	n	soma
0		10	0
1	3	20	10
2	2	30	30
3	1	40	60
4	0	50	100

V

0	1	2	3

V

10			
----	--	--	--

V

10	20		
----	----	--	--

V

10	20	30	
----	----	----	--

V

10	20	30	40
----	----	----	----

V

10	20	30	40
----	----	----	----

r
25

b.

<i>i</i>	<i>j</i>	<i>n</i>	<i>soma</i>
0	3	10	10
1	2	20	20
2	1	30	30
3	0	40	60
4	-1	50	160

V

0	1	2	3

V

10			
----	--	--	--

V

10	20		
----	----	--	--

V

10	20	30	
----	----	----	--

V

10	20	30	40
----	----	----	----

V

10	20	30	40
----	----	----	----

<i>r</i>
40

c.

<i>i</i>	<i>j</i>	<i>n</i>	<i>soma</i>
0		10	0
1	3	20	10
2	2	30	30
3	1	40	60
4	0	50	100

V

0	1	2	3

V

			10
--	--	--	----

V

		20	10
--	--	----	----

V

		20	10
--	--	----	----

V

	30	20	10
--	----	----	----

V

40	30	20	10
----	----	----	----

<i>r</i>
Erro! Divisão por zero

✓ d.

<i>i</i>	<i>j</i>	<i>n</i>	<i>soma</i>
0		10	0
1	3	20	10
2	2	30	30
3	1	40	60
4	0	50	100

V

0	1	2	3

V

--	--	--	--

V

			10
--	--	--	----

V

		20	10
--	--	----	----

V

	30	20	10
--	----	----	----

V

40	30	20	10
----	----	----	----

<i>r</i>
25

e.

<i>i</i>	<i>j</i>	<i>n</i>	<i>soma</i>
1	3	10	0
2	2	20	10
3	1	30	30
4	0	40	60
5	0	50	100

V

0	1	2	3

V

			20
--	--	--	----

V

			20
--	--	--	----

V

		30	20
--	--	----	----

V

	40	30	20
--	----	----	----

V

50	40	30	20
----	----	----	----

<i>r</i>
20

da
resposta:

Resposta: D

Comentário: Nas linhas 6 a 8 do algoritmo as variáveis i , n e $soma$ são inicializadas com os valores 0, -10 e 0, respectivamente.

i	j	n	$soma$
0		10	0

Na linha 9 inicia uma laço de repetição, controlado pela variável j e o algoritmo repetirá as linhas 10 a 13 4 vezes. Para j de 3 até 1 passo -1 execute os comandos das linhas 10 a 13.

10. $v[j] \leftarrow n$
11. $n \leftarrow n + 10$
12. $soma \leftarrow soma + v[j]$
13. $i \leftarrow i + 1$

No comando da linha 10, ao vetor v , índice j , será atribuído o valor de n . Na primeira vez em que esta atribuição é feita, n vale 10. A variável j foi iniciada com 3 e será decrementado em -1, então o vetor será preenchido da última posição para a primeira.

A variável i foi iniciada com 0 na linha 13, mas dentro do laço, a cada iteração, ela é incrementada de -1 em 1 e, como o laço é executado 4 vezes, ao término da execução do laço, ela está valendo 4.

A variável n foi iniciada na linha 11 e, dentro do laço de repetição, tem o seu valor incrementado em 10 posições. Primeiro o valor de n é atribuído ao vetor v na posição j . $v[j] \leftarrow n$, em seguida n é incrementado em +10. Neste caso, ao término de cada iteração do laço, a variável n é acrescentado 10 e este valor será atribuído ao vetor na próxima iteração.

A variável $soma$ tem o seu valor acumulado com o valor atribuído à variável $v[j]$. Neste caso, a variável $soma$ guarda o somatório de todos os elementos do vetor a cada iteração. Como o vetor guarda os valores 40, 30, 20, 10, ao término da execução, a variável $soma$ vale 100.

Isto posto, a alternativa que representa corretamente o valor de cada variável durante a execução do algoritmo é a alternativa D.

As alternativas A e B estão incorretas porque representam o vetor preenchido a partir da posição 0, enquanto o algoritmo o preenche a partir da posição 3.

A alternativa B está incorreta também porque a variável $soma$ está com os valores computados errados.

A alternativa C está incorreta porque a variável r está apresentando o resultado da divisão de $soma$ por j , como j é 0, a divisão por zero promove uma falha. A instrução correta é a divisão de $soma$ por i , $soma$ vale 100 e i vale 4, logo $100/4 = 25$.

A alternativa E está incorreta porque o resultado considera a execução da linha 10 invertida com a linha -11. Como o algoritmo é sequencial, primeiro será atribuído o valor da variável n ao vetor $v[j]$ e em seguida, o valor de n será incrementado em +10.

Pergunta 2

0,25 em 0,25 pontos



Considere v um vetor de inteiros, contendo a sequência de valores {7, 9, 15, 21, 45, 53, 75, 81} e o algoritmo de busca dado abaixo, o qual recebe como argumento dois valores, sendo que n corresponde ao tamanho do vetor e x o valor a ser pesquisado. Analise o algoritmo e as asserções, verificando quais estão corretas.

```
1.      Algoritmo "Questao 2 Unidade 4"
2.      Var
3.      v[0..9] : inteiro
4.
5.      // função para buscar o valor x no vetor
6.      de tamanho n.
7.      funcao busca(n, x : inteiro) : inteiro
8.      var
9.      e, d, m : inteiro
10.     inicio
11.     e <- 0
12.     d <- n-1
13.     enquanto (e <= d) faca
14.         m <- ((e + d) div 2)
15.         se (v[m] = x) então
16.             retorne m
17.         senão
18.             se (v[m] > x) então
19.                 d <- m-1
20.             senão
21.                 e <- m+1
22.         fimse
23.     fimse
24.     fimenquanto
25.     retorne -1
26. fimfuncao
27. Inicio
28.     r <- busca(8, 53)
29.     s <- busca(8, 11)
30.     t <- busca(8, 7)
31.     u <- busca(8, 81)
32. Fimalgoritmo
```

- I. A variável r vale 5 porque uma chamada à função realizada pelo comando $r \leftarrow busca(8, 53)$ na linha -27 do bloco principal do algoritmo retornará 5.
- II. O número máximo de iterações que o algoritmo irá realizar quando o valor pesquisado não existir no vetor $v[]$ é 4.
- III. Este algoritmo funciona apenas quando os dados estão classificados no vetor.
- IV. Ao término da execução do algoritmo, a variável t vale -1.

Resposta Selecionada: ☒ d. As asserções I, II e III estão corretas.

Respostas:

- a. As asserções I e II estão corretas.
- b. As asserções III e IV estão corretas.
- c. As asserções II e III estão corretas.
- ☒ d. As asserções I, II e III estão corretas.
- e. As asserções II, III e IV estão corretas.

Comentário da resposta:

Resposta: D

Comentário: O algoritmo implementa a busca binária. Classicamente, a primeira posição do vetor é zero e a última é $n-1$, onde n é o tamanho do vetor. Este algoritmo considera a variável e como a posição mais à esquerda do vetor, inicialmente 0 e d a posição mais à direita do vetor, inicialmente, o tamanho do vetor menos 1, ou $n-1$. A variável m corresponde a posição mais central do vetor, ou seja, o quociente da divisão $(e+d)/2$. Se o valor pesquisado é encontrado, a função `busca()` retorna o índice do vetor do valor pesquisado, ou -1 caso contrário.

A rmativa I. Correta

No comando `r <- busca(8, 53)` da linha 28 do bloco principal, a variável r recebe a posição onde o valor 53 é encontrado, neste caso, a posição 5.

A rmativa II. Correta

O pior cenário dos dados para esta busca é quando o valor pesquisado não é encontrado, ou quando está na primeira ou na última posição do vetor. O pior cenário está associado ao cenário em que o algoritmo requer mais tempo de processamento. Como o algoritmo testa a posição mais central do vetor e , o valor é encontrado quando o valor está posicionado na posição mais central, o algoritmo apresenta desempenho similar para localizar os elementos das posições extremas ou quando o valor pesquisado não é encontrado.

Para buscar um valor que não existe, menor que o valor armazenado na posição central do vetor, na primeira iteração, $e=0$, $d=7$ e $m=3$ e o vetor é representado da forma como segue:

V	7	9	15	21	45	53	75	81
	0	1	2	3	4	5	6	7
	e			m				d

Na linha 14, o algoritmo testa se o valor pesquisado (11) está em $v[m]$. Na primeira iteração o resultado é falso e o bloco `senão` será executado. Neste bloco, há uma outra condição a ser verificada, ou seja, se o valor armazenado em $v[m]$ é maior que x (x é o valor sendo pesquisado), sendo o resultado menor, e o bloco `senão` será executado.

Na prática, o algoritmo busca o elemento entre as posições marcadas entre os valores armazenados nas variáveis e e d . Como o valor é menor que o elemento que está em $v[m]$ e os dados do vetor estão ordenados, deduz-se que o valor está em alguma posição a esquerda do vetor e as posições à direita são ignoradas a partir desta iteração.

V	7	9	15	21	45	53	75	81
	0	1	2	3	4	5	6	7
	e	m	d					

O algoritmo recebe o valor das variáveis d e m e repete o processo. Nesta iteração, a posição mais central é a posição 1. Como a posição 1 também não guarda o elemento pesquisado, o algoritmo vai verificar se o valor pesquisado (11) é maior que o valor encontrado na posição central com resposta VERDADEIRO. Neste caso a variável e será atualizada para $m+1$ e as posições à esquerda da posição m serão ignoradas conforme ilustrado abaixo:

V	7	9	15	21	45	53	75	81
	0	1	2	3	4	5	6	7
			e					
			d					
			m					

Na terceira iteração, as variáveis e , d e m são iguais a 2. O algoritmo testa se o valor pesquisado (11) está em $v[m]$, com resultado FALSO, visto que $v[m] = 15$. O algoritmo segue verificando se $v[m] > 11$, resultando FALSO. Neste caso, a variável d recebe $m-1$, ou seja, 1 e, quando a condição do laço enquanto for verificada, o valor da variável e é maior que o valor da variável d , significa que o valor pesquisado não está no vetor, interrompendo o ciclo do laço enquanto e retornando -1.

V	7	9	15	21	45	53	75	81
	0	1	2	3	4	5	6	7
		d	e					

Como o vetor tem tamanho par, os lados esquerdos e direito têm número de elementos diferentes, excluindo a posição mais central do vetor. Caso o valor pesquisado estivesse do lado direito do vetor, haveria uma iteração a mais.

A afirmativa III. Correta.

Este algoritmo funciona apenas quando os dados estão classificados e, não importa se em ordem crescente ou decrescente. Isto devido às comparações, se maior que o valor mais central do vetor.

A afirmativa IV. Incorreta.

Conforme verificado, o algoritmo sempre retorna o índice do vetor, quando o valor existe. O valor 7 existe na posição 0 do vetor, logo a variável t recebe 0.

Pergunta 3

0,25 em 0,25 pontos



Considere o algoritmo:

```
1.      Algoritmo "Questao 2 Unidade 4"
2.      Var
3.      i, j, aux, v[0..9] : inteiro
4.
5.      Inicio
6.      para i de 0 ate 9 passo 1 faca
7.          v[i] <- i * 10
8.      fimpara
9.
10.     i <- 0
11.     j <- 9
12.
13.     enquanto (i < 5) faca
14.         aux <- v[i]
15.         v[i] <- v[j]
16.         v[j] <- aux
17.         i <- i+1
18.         j <- j-1
19.     fimenquanto
20.     Fimlgoritmo
```

Quais os valores armazenados no vetor v em sua respectiva sequência após a execução do algoritmo?

Resposta Selecionada: ☒ c. 90, 80, 70, 60, 50, 40, 30, 20, 10, 0.

Respostas: a. 9, 8, 7, 6, 5, 4, 3, 2, 1, 0.

b. 0, 1, 2, 3, 4, 5, 6, 7, 8, 9.

☒ c. 90, 80, 70, 60, 50, 40, 30, 20, 10, 0.

d. 90, 80, 70, 60, 50, 0, 10, 20, 30, 40.

e. 0, 10, 20, 30, 40, 50, 60, 70, 80, 90.

Comentário Resposta: C

da
resposta: Comentário: Após a execução do laço para nas linhas 6 a 8, o vetor v armazenará os seguintes dados:


```

6. para i de 0 ate 9 passo 1 faca
7.   v[i] <- i * 10
8. fimpara
v[0] = 0
v[1] = 10
v[2] = 20
v[3] = 30
v[4] = 40
v[5] = 50
v[6] = 60
v[7] = 70
v[8] = 80
v[9] = 90

```

O laço enquanto terá 5 iterações e fará a troca dos valores que estão nas primeiras 5 posições com os valores que estão nas 5 últimas posições.

```

enquanto (i < 5) faca
  aux <- v[i]
  v[i] <- v[j]
  v[j] <- aux
  i <- i+1
  j <- j-1
fimenquanto

```

Em cada iteração, as variáveis assumem os seguintes valores:

aux <- v[i]	v[i] <- v[j]	v[j] <- aux			
1a. iteracao, i=0, j=9,	aux ← v[0] aux = 0	v[0] ← v[9] ⇔ v[0] = 90	v[0] = 0	v[9] ← aux ⇔ v[9] = 0	
2a. iteracao, i=1, j=8,	aux ← v[1] aux = 10	v[1] ← v[8] ⇔ v[1] = 80	v[1] = 10	v[8] ← aux ⇔ v[8] = 10	
3a. iteracao, i=2, j=7,	aux ← v[2] aux = 20	v[2] ← v[7] ⇔ v[2] = 70	v[2] = 20	v[7] ← aux ⇔ v[7] = 20	
4a. iteracao, i=3, j=6,	aux ← v[3] aux = 30	v[3] ← v[6] ⇔ v[3] = 60	v[3] = 30	v[6] ← aux ⇔ v[6] = 30	
5a. iteracao, i=4, j=5,	aux ← v[4] aux = 40	v[4] ← v[5] ⇔ v[4] = 50	v[4] = 40	v[5] ← aux ⇔ v[5] = 40	

o término do algoritmo, o vetor $v[]$ estará armazenando os seguintes valores em suas posições:

```

v[0] = 90
v[1] = 80
v[2] = 70
v[3] = 60
v[4] = 50
v[5] = 40
v[6] = 30
v[7] = 20
v[8] = 10
v[9] = 0

```

Diante do exposto, conclui-se que:

A alternativa A está errada porque apenas a posição 0 do vetor está correta. Exceto na posição que armazena o valor zero, o vetor não armazena valores menores do que 10.

A alternativa B está errada porque apresenta os valores errados e numa sequência errada.

A alternativa D está errada porque metade do vetor está com os valores corretos, mas em posições erradas.

A alternativa E está errada porque apresenta o vetor após o preenchimento, mas não considera as trocas realizadas dentro do laço enquanto.

Isto posto, a única alternativa que responde corretamente a questão é C.

Pergunta 4

0,25 em 0,25 pontos



Considere o vetor `v[]` de tamanho 10, declarado na área de declaração de variáveis e os comandos de atribuição dados a seguir:

```
Var
  v : vetor[0..9] de inteiro
Inicio
  v[10] <- 65
  v[7] <- 5 div 2
  v[] <- 8
  v[1,3] <- 30
Fimalgoritmo
```

I. O comando `v[10] <- 65` resultará em erro devido à tentativa de acessar um índice inexistente do vetor.

II. O comando `v[5] <- 5 div 2` resultará em erro devido a tentativa de armazenar um número real num endereço de memória declarado para armazenar apenas inteiros.

III. O comando `v[] <- 8` armazena o valor inteiro 8 em uma posição aleatória do vetor `v[]`.

IV. O comando `v[1,3] <- 30` funciona corretamente e armazenará na única linha do vetor, na coluna 3, o valor 30.

Assinale a alternativa correta.

Resposta Selecionada: ☒ a. Apenas a alternativa I.

Respostas: ☒ a. Apenas a alternativa I.
☐ b. Apenas a alternativa II.
☐ c. Apenas a alternativa III.

d. Apenas a alternativa IV.

e. Nenhuma alternativa está correta.

Comentário da resposta:

Resposta: A
Comentário:

A alternativa I. CORRETA.

Embora o vetor seja de tamanho 10, as posições variam de 0 a 9, portanto se n é o tamanho do vetor, os índices válidos variam de 0 a $n-1$. Portanto, a tentativa de acessar um índice inexistente no vetor resultará em erro.

A alternativa II. INCORRETA

O vetor `v[]` foi declarado para armazenar valores inteiros. O comando `5 div 2` retorna o quociente da divisão ($5/2$) que é um inteiro. Portanto, o comando será processado corretamente.

A alternativa III. INCORRETA

O comando `v[] <- 8` retorna um erro porque é obrigatório informar o índice do vetor a armazenar o valor -8. Em algumas linguagens de programação é possível declarar e atribuir ao mesmo tempo e, neste caso, os valores atribuídos ao vetor devem estar na notação de conjunto, da seguinte forma:

```
int v1[] = {8}
int v2[] = {10, 30, 40, 50}
```

Desta forma, `v1[]` é um vetor com um único elemento e `v2[]` é um vetor com 4 elementos. A sequência dos elementos estabelece seus índices e o tamanho do conjunto é o tamanho do vetor. Esta notação não funciona no pseudocódigo interpretado pelo VisualG e, por este motivo, não é aplicado.

A alternativa IV. INCORRETA

O comando `v[1,3] <- 30` não funciona corretamente porque o vetor `v[]` não foi declarado como matriz.

Pergunta 5

0,25 em 0,25 pontos



O algoritmo a seguir atribui 50 valores numéricos inteiros numa matriz 10×5 , calcula e armazena numa segunda matriz os 50 valores da primeira matriz multiplicados por 10. Escolha o trecho de código que mostra corretamente os valores armazenados nas matrizes `num[][]` e `mult[][]`.

Algoritmo "Unidade 4 - Questão 5"

Var

num : vetor[0..9,0..4] de inteiro
mult: vetor[0..9,0..4] de inteiro
i, j, linha, coluna : inteiro

Inicio

para i de 0 ate 9 passo 1 faca
 para j de 0 ate 4 passo 1 faca
 num[i][j] = i + j
 mult[i][j] = num[i][j] * 10;
 fimpara
fimpara
// mostrar os elementos das matrizes

Fimalgoritmo

I. enquanto (i<9)
 repita
 escreva("num[" , i, "]" , j, "]=", num[i][j])
 escreva("mult[" , i, "]" , j, "]=", mult[i][j])
 j<-j+1
 ate (j=4)
 i<-i+1
 fimenquanto

II. para j de 0 ate 9 passo 1 faca
 para i de 0 ate 4 passo 1 faca
 escreva("num[" , j, "]" , i, "]=", num[j][i])
 escreva("mult[" , j, "]" , i, "]=", mult[j][i])
 fimpara
 fimpara

III. i<-0
 enquanto (i<9)
 j<-0
 repita
 escreva("num[" , i, "]" , j, "]=", num[i][j])
 escreva("mult[" , i, "]" , j, "]=", mult[i][j])
 j<-j+1
 ate (j=4)
 i<-i+1
 fimenquanto

IV. i<-0
 faca
 j<-0
 faca
 escreva("num[" , i, "]" , j, "]=", num[i][j])
 escreva("mult[" , i, "]" , j, "]=", mult[i][j])
 enquanto (i<10)
 j<-j+1
 i<-i+1
 enquanto (j<4)

Estão corretos e completam o algoritmo os seguintes trechos:

Resposta Seleccionada: ☒ b. II e III, apenas.

Respostas: ☐ a. I e II, apenas.

☒ b. II e III, apenas.

☐ c. III e IV, apenas.

☐ d. I e III, apenas.

☐ e. II e IV, apenas.

Comentário da resposta:

Resposta: B
Comentário:

I. Incorreto porque as variáveis i e j não foram inicializadas. Ao término do laço para ... mpara, as variáveis i e j estarão valendo, respectivamente, 9 e 4 e a verificação enquanto (i<9) resultará falso.

II. Correto porque as variáveis i e j foram inicializadas corretamente. A variável i foi inicializada com 0 antes de entrar no primeiro laço enquanto e a variável j foi inicializada com 0 antes do segundo laço.

III. Correto porque está seguindo a mesma estrutura usada para preencher o vetor.

IV. Incorreto porque utiliza o laço de repetição com teste nm e, neste laço, a condição não é verificada antes de iniciar o bloco de repetição. Observe que a variável que controla o laço mais interno está sendo incrementada fora do bloco de repetição e, por este motivo, o valor nunca é alterado, impactando num laço in nito.

Pergunta 6

0,25 em 0,25 pontos



Dada a matriz notas representada e o vetor alunos, o objetivo do algoritmo é transpor os dados do vetor alunos na primeira coluna da matriz notas, preencher as notas de cada bimestre nas colunas 1 a 4 para cada aluno e, na última coluna da matriz notas, preencher com a média ponderada dos alunos, de acordo com os seguintes pesos (B1=1, B2=2, B3=3, B4=4). Algumas linhas dos algoritmos precisam ser preenchidas, então escolha a alternativa que completa corretamente o algoritmo.

aluno	001	002	003	004	005	006	007
-------	-----	-----	-----	-----	-----	-----	-----

Aluno	N1	N2	N3	N4	Média
001	10.0	10.0	8.0	8.0	$=10 \cdot 0,1 + 10 \cdot 0,2 + 8 \cdot 0,3 + 8 \cdot 0,4$
002					
003					
004					
005					
006					
007					

Algoritmo "Unidade 4 – Questão 6"

Var

(1) _____

media_turma, soma, notas: vetor[0..7,0..4] de real
i, j: inteiro

Inicio

//preencher o vetor aluno com os códigos dos alunos.

para i de 0 ate 7 passo 1 faca

escreva("Digite o código do aluno")

leia(aluno[i])

fimpara

para i de 0 ate 7 passo 1 faca

 soma <- 0

 //preencher os códigos dos alunos na 1ª. coluna

 (2) _____

 (3) _____

leia(notas[i,j])

 (4) _____

fimpara

 (5) _____

fimpara

 // mostrar os elementos das matrizes

para i de 1 ate 7 passo 1 faca

para j de 1 ate 4 passo 1 faca

 (6) _____

fimpara

fimpara

Fimalgoritmo

Resposta Selecionada:

(1) aluno: vetor[0..6] de inteiro

(2) notas[i,0] <- aluno[i]

(3) para j de 1 ate 4 passo 1 faca

(4) soma <- (soma + (notas[i,j]*(i/10)))

(5) notas[i,5] <- soma/i

✓ a. (6) escreva(notas[i,j])

Respostas:

(1) aluno: vetor[0..6] de inteiro

(2) notas[i,0] <- aluno[i]

(3) para j de 1 ate 4 passo 1 faca

(4) soma <- (soma + (notas[i,j]*(i/10)))

(5) notas[i,5] <- soma/i

✓ a. (6) escreva(notas[i,j])

- (1) aluno: vetor[0..6] de inteiro
- (2) notas[i,j] <- aluno[i]
- (3) para j de 0 ate 4 passo 1 faca
- (4) soma <- (soma + (notas[i,j]*(i/10)))
- (5) notas[i,j] <- soma/i
- (6) escreva(notas[i,j])

b.

- (1) aluno: vetor[1..7] de inteiro
- (2) notas[0,i] <- aluno[i]
- (3) para j de 0 ate 4 passo 1 faca
- (4) soma <- (soma + (notas[i,j]*(i/10)))
- (5) notas[i,5] <- soma/i
- c. (6) escreva(notas[i,j])

- (1) aluno: vetor[0..6] de inteiro
- (2) notas[j,i] <- aluno[j]
- (3) para j de 1 ate 4 passo 1 faca
- (4) soma <- (notas[j,i]*(j/10))
- (5) notas[i,5] <- soma/i
- d. (6) escreva(notas[j,i])

- (1) aluno: vetor[0..7] de inteiro
- (2) notas[i,0] <- aluno[i]
- (3) para j de 1 ate 7 passo 1 faca
- (4) soma <- (notas[i,j]/4))
- (5) notas[i,5] <- soma/i
- e. (6) escreva(notas[i,j])

Comentário da
resposta:

Resposta: A

Comentário: A alternativa A completa corretamente o algoritmo, conforme pode ser visto a seguir.

Algoritmo "Unidade 4 - Questão 6"

Var

aluno: vetor[0..7] de inteiro
media_turma, soma, notas: vetor[0..7,0..4] de real
i, j: inteiro

Inicio

//preencher o vetor aluno com os códigos dos alunos.

para i de 0 ate 7 passo 1 faca

escreva("Digite o código do aluno")

leia(aluno[i])

fimpara

para i de 0 ate 7 passo 1 faca

soma <- 0

//preencher os códigos dos alunos na 1ª. coluna

notas[i,0] <- aluno[i]

para j de 1 ate 4 passo 1 faca

leia(notas[i,j])

soma <- (soma + (notas[i,j]*(i/10)))

fimpara

notas[i,5] <- soma/i

fimpara

// mostrar os elementos das matrizes

para i de 1 ate 7 passo 1 faca

para j de 1 ate 4 passo 1 faca

escreva(notas[i,j])

fimpara

fimpara

Fimalgoritmo

A alternativa B está errada porque as linhas (2)) notas[i,j] <- aluno[i] e (5) notas[i,j] <- soma/i

estão erradas. Na linha 2, a ideia é transpor o vetor para uma coluna da matriz e todos os códigos serão armazenados na coluna 0 da matriz, então a linha poderá variar de 1 a 7, mas a coluna é (sempre a coluna 0). A linha 5 está errada porque ao referenciar a matriz notas com os índices i,j, linha e coluna são dados dinâmicos, mas neste caso, a coluna que a nota será atribuída é a coluna 5.

A alternativa C está errada porque o vetor está sendo declarado com as posições de 1 a 7, mas o algoritmo faz referência as posições de 0 a 6. Nas linguagens de programação, o primeiro índice do vetor é sempre o zero.

A alternativa D está errada porque o comando da linha 2 está inconsistente (2) notas[j,i] <- aluno[j] . Há dois laços de repetição no algoritmo. O laço externo é controlado pela variável i e o interno pela variável j. A cada iteração do laço externo, há 4 iterações do laço i. Desta forma, inverter as posições na referência ao vetor pode ser algo muito grave.

A alternativa E está errada porque a linha 4 não está acumulando a soma, com isso o resultado fica errado.

Pergunta 7

0,25 em 0,25 pontos



Antes de executar o procedimento p() no VisualG será necessário um vetor de números inteiros declarado como variável, conforme o exemplo a seguir:

Algoritmo "Unidade 4 - questao 7"

Var

k : inteiro
vet : vetor[0..11] de inteiro

No procedimento, a parâmetro tam representa o tamanho do vetor.

```
procedimento p(tam : inteiro)
var
  i, j, min, x : inteiro
inicio
  para i de 1 ate tam-1 passo 1 faca
    min <- i
    para j de (i+1) ate tam passo 1 faca
      se (vet[j]<vet[min]) entao
        min <- j
    fimse
  fimpara
  aux <- vet[i]
  vet[i] <- vet[min]
  vet[min] <- aux
fimpara
fimprocedimento
```

Considerando as entradas abaixo,

```
Inicio
vet[0] <- 01
vet[1] <- 91
vet[2] <- 10
vet[3] <- 27
vet[4] <- 40
vet[5] <- 76
vet[6] <- 62
vet[7] <- 85
vet[8] <- 18
vet[9] <- 37
vet[10] <- 56
vet[11] <- 99
```

Assinale a opção que representa corretamente o funcionamento do algoritmo.

Resposta Selecionada: ☒ a. O algoritmo classifica os dados do vetor em ordem crescente.

Respostas: ☒ a. O algoritmo classifica os dados do vetor em ordem crescente.
b.
O propósito do algoritmo é reorganizar os dados do vetor de tal forma que seus elementos, ao final, estejam ordenados de forma crescente.
c. O algoritmo classifica os dados do vetor em ordem decrescente.
d. O algoritmo pesquisa o menor elemento do vetor.
e. O algoritmo pesquisa o maior elemento do vetor.

f. O algoritmo soma os elementos do vetor.

Comentário da resposta: Resposta: A.
Comentário: O procedimento $p()$ implementa o algoritmo de ordenação conhecido como selectionsort. A lógica deste algoritmo é encontrar o menor elemento do vetor e, em seguida, troca o elemento na posição do menor com o elemento que está na posição 0. A posição zero deve armazenar o menor elemento do vetor. O algoritmo selecionará o próximo menor e, quando encontrar, o armazenará na posição 1. Vai procurar o próximo menor elemento do vetor e, ao encontrar, vai trocar com o elemento que está na posição 3.

A alternativa A está correta porque os dados são armazenados no vetor em ordem crescente.

A alternativa B está errada porque os dados são armazenados em ordem crescente.

A alternativa C está errada porque, apesar do algoritmo fazer uma busca pelo menor, esta busca faz parte do processo mas não é o objetivo.

A alternativa D está errada porque o algoritmo não pesquisa o maior elemento do vetor.

A alternativa E está errada porque o algoritmo não soma os elementos do vetor.

Pergunta 8

0,25 em 0,25 pontos



Considere uma matriz quadrada conforme exemplificada abaixo, na qual a diagonal principal (destacada em amarelo) está preenchida com o número da linha e, a porção triangular inferior (destacadas em verde) está preenchida com números sequenciais.

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

A porção triangular superior é simétrica e deve ser preenchida por processamento de software. A regra consiste em verificar se há valor numa linha e coluna, como por exemplo, na linha 2, coluna 1 está armazenado o valor 3. Então o procedimento deve escrever na linha 1, coluna 2 o mesmo valor.

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

Analise os códigos de I a IV e escolha a opção que identifique corretamente os módulos de acordo com o propósito.

Algoritmo "Unidade 4 - questao 8"

Var

linha, coluna, k, aux : inteiro

mat : vetor[0..5,0..5] de inteiro

I.

procedimento _____()

var

linha, coluna : inteiro

inicio

para linha de 0 ate 5 passo 1 faca

escreval(" ")

para coluna de 0 ate 5 passo 1 faca

escreva(" ", mat[linha,coluna], " ")

fimpara

fimpara

fimprocedimento

II.

procedimento _____()

var

x, y, linha, coluna : inteiro

inicio

para linha de 0 ate 5 passo 1 faca

para coluna de 0 ate 5 passo 1 faca

se (coluna<linha) entao

mat[linha,coluna] <- linha+coluna

fimse

fimpara

fimpara

fimprocedimento

III.

procedimento _____()

var

x, y, linha, coluna : inteiro

inicio

para linha de 0 ate 5 passo 1 faca

para coluna de 0 ate 5 passo 1 faca

se (linha=coluna) entao

mat[coluna,linha] <- linha+1

fimse

fimpara

fimpara

fimprocedimento

IV.

Resposta ☒ a.

Selecionada: Inicialização do algoritmo e área de declaração de variáveis, escreverMatriz(), preencherMatriz, preencherDiagonalPrincipal.

Respostas: ☒ a.

Inicialização do algoritmo e área de declaração de variáveis, escreverMatriz(), preencherMatriz, preencherDiagonalPrincipal.

b.

Inicialização do algoritmo e área de declaração de variáveis, preencherMatriz(),preencherDiagonalPrincipal, escreverMatriz.

c.

Inicialização do algoritmo e área de declaração de variáveis, preencherMatriz(),preencherDiagonalPrincipal, escreverMatriz.

d.

escreverMatriz(), preencherMatriz, preencherDiagonalPrincipal, Inicialização do algoritmo e área de declaração de variáveis

e.

pesquisa(), preencherMatriz, preencherDiagonalPrincipal, Inicialização do algoritmo e área de declaração de variáveis

Comentário da resposta:

Resposta: A.

Comentário: A alternativa que responde corretamente a questão é a A. Os códigos devidamente identificados seguem abaixo,

I. O trecho de código consiste da identificação da variável e a área de declaração de variáveis.

```
Algoritmo "Unidade 4 - questao 8"
Var
    linha, coluna, k, aux : inteiro
    mat : vetor[0..5,0..5] de inteiro

procedimento escreverMatriz()
var
    linha, coluna : inteiro
inicio
    para linha de 0 ate 5 passo 1 faca
        escreval(" ")
        para coluna de 0 ate 5 passo 1 faca
            escreva(" ", mat[linha,coluna], " ")
        fimpara
    fimpara
fimprocedimento

procedimento preencherMatriz()
var
    x, y, linha, coluna : inteiro
inicio
    para linha de 0 ate 5 passo 1 faca
        para coluna de 0 ate 5 passo 1 faca
            se (coluna<linha) entao
                mat[linha,coluna] <- linha+coluna
            fimse
        fimpara
    fimpara
fimprocedimento

procedimento preencherDiagonalPrincipal()
var
    x, y, linha, coluna : inteiro
inicio
    para linha de 0 ate 5 passo 1 faca
        para coluna de 0 ate 5 passo 1 faca
            se (linha=coluna) entao
                mat[coluna,linha] <- linha+1
            fimse
        fimpara
    fimpara
fimprocedimento
```

Pergunta 9

0,25 em 0,25 pontos



Considere uma matriz quadrada, na qual na última coluna estão preenchidos com os totalizadores de cada linha conforme exempli cada abaixo.

	0	1	2	3	4
0	10	20	30	40	100
1	20	40	60	80	200
2	30	60	90	120	300
3	40	80	120	160	400
4	50	100	150	200	500
5	60	70	80	90	300

Assuma que a matriz já exista e esteja parcialmente preenchida, o algoritmo deve somar cada linha da matriz e preencher o resultado na coluna 5. Considerando que nesse algoritmo há erros de lógica de programação, que devem ser corrigidos, assinale a opção correta no que se refere as adequações necessárias.

```
1.  procedimento totalizar()
2.  var
3.    x, y : inteiro
4.  inicio
5.    para linha de 1 ate 5 passo 1 faca
6.      para coluna de 0 ate 4 passo 1 faca
7.        mat[linha,coluna] <- mat[linha,5] + mat[linha,coluna]
8.      fimpara
9.    fimpara
10. fimprocedimento
```

Assinale a alternativa que corrige o procedimento totalizar()

Resposta: ☒ a.

Selecionada: Na linha 3 devem renomear as variáveis para linha e coluna. Na linha 5 deve ser corrigido da seguinte forma para linha de 0 ate 4 passo 1 faca e a linha linha 7, do seguinte modo, mat[linha,5] <- mat[linha,5] + mat[linha,coluna].

Respostas: ☒ a.

Na linha 3 devem renomear as variáveis para linha e coluna. Na linha 5 deve ser corrigido da seguinte forma para linha de 0 ate 4 passo 1 faca e a linha linha 7, do seguinte modo, mat[linha,5] <- mat[linha,5] + mat[linha,coluna].

b.

Na linha 5 deve ser corrigido da seguinte forma para linha de 0 ate 4 passo 1 faça e a linha linha 6, do seguinte modo, para coluna de 0 ate 4 passo 1 faça.

c.

Na linha 3 deve ser corrigido da seguinte forma linha, coluna : caractere e na linha e a linha 7, do seguinte modo, `mat[linha,coluna] <- mat[linha,5] + mat[linha,coluna]`. Na linha 1 deve ser corrigido da seguinte forma função `totalizar()` : inteiro, na linha 10 deve ser incluído o comando `retorne x` e na linha 11 do seguinte modo `mfuncao`.

d.

Na linha 1 deve ser corrigido da seguinte forma função `totalizar()` : inteiro e na linha 10 do seguinte modo `mfuncao`

Comentário da resposta:

Resposta: A

Comentário: A alternativa que responde corretamente a questão é a alternativa A.

A alternativa B está errada porque está incompleta, uma vez que não identificou o erro da linha 3.

A alternativa C está errada porque sugere que as variáveis sejam do tipo caractere e, este tipo destoa do propósito do algoritmo.

A alternativa D está errada porque sugere que o procedimento seja implementado como uma função, lembrando de adicionar o comando `retorne`

A alternativa E está errada porque além de sugerir a implementação como função, não aponta os erros já comentados referentes às linhas 3, 5 e 7.

A alternativa E está errada porque não aponta os erros já comentados referentes às linhas 3, 5 e 7.

Versão corrigida do algoritmo:

```
1. procedimento totalizar()  
2. var  
3.     linha, coluna : inteiro  
4. inicio  
5.     para linha de 0 ate 5 passo 1 faça  
6.         para coluna de 0 ate 3 passo 1 faça  
7.             mat[linha,5] <- mat[linha,5] +  
8.             mat[linha,coluna]  
9.         fimpara  
10.    fimpara  
11. fimprocedimento
```

Pergunta 10

0,25 em 0,25 pontos



Analise o algoritmo abaixo e as alternativas. Cada alternativa está associada a um comentário no código. Assinale a alternativa correta:

Algoritmo " Unidade 4 - Questao 10"

```
i : inteiro
v : vetor[0..5] de inteiro
w : vetor[0..3] de inteiro
k : vetor[0..9] de inteiro
Inicio

// I.
aleatorio on
  para i de 0 ate 5 passo 1 faca
    se (i<4) entao
      leia(v[i])
      leia(w[i])
    senao
      leia(v[i])
    fimse
  fimpara
aleatorio off

// II.
para i de 0 ate 5 passo 1 faca
  se (i<=3) entao
    k[i] <- v[i]
    k[i+6] <- w[i]
  senao
    k[i] <- v[i]
  fimse
fimpara

// III.
para i de 0 ate 5 passo 1 faca
  escreval("v[",i,"]= ", v[i])
fimpara
para i de 0 ate 3 passo 1 faca
  escreval("w[",i,"]= ", w[i])
fimpara
para i de 0 ate 9 passo 1 faca
  escreval("k[",i,"]= ", k[i])
fimpara
Fimalgoritmo
```

I. A respeito do bloco de código do comentário I, é correto afirmar que o algoritmo preencherá as posições 0 a 4, inclusive, de ambos os vetores simultaneamente e as últimas posições a serem preenchidas com valores aleatórios são as posições 3 e 4 do vetor v .

II. A respeito do bloco de código do comentário II, é correto afirmar que a cada iteração do laço, dois dados são atribuídos ao vetor k , até que faltem apenas duas posições.

III. A respeito do bloco de código do comentário III, é correto afirmar que os três laços de repetição poderiam ser reduzidos a um, mesmo sendo para mostrar conteúdos de 3 vetores com tamanhos diferentes.

IV. As variáveis i , v , w , k são variáveis locais e acessíveis em todos os módulos.

Resposta Seleccionada:  b. V - V - V - F.

✓ b. $V - V - V - F$.

d. $V - F - V - F$.

Comentário Resposta: B.

```
//1.
aleatorio on
  para i de 0 ate 5 passo 1 faca
    se (i<4) entao
      leia(v[i])
      leia(w[i])
    senao
      leia(v[i])
    fimse
  fimpara
aleatorio off
```

```
//II.
para i de 0 ate 5 passo 1
faca
    se (i<=3) entao
        k[i] <- v[i]
        k[i+6] <- w[i]
    senao
        k[i] <- v[i]
    fimse
fimpara
```

v: vetor[0..5] de inteiro
w: vetor[0..3] de inteiro
k: vetor[0..9] de inteiro

k

--	--	--	--	--	--	--	--	--	--

v w

--	--	--	--	--	--	--	--	--	--

III. Correta

Os s três laços de repetição poderiam ser reduzidos a um, mesmo sendo para mostrar conteúdos de 3 vetores com tamanhos diferentes.

```
// III.  
para i de 0 ate 5 passo 1 faca  
    escreval("k[,i,]= ", k[i])  
    se (i<=3) então  
        escreval("k[,i,]= ", k[i])  
        escreval("w[,i,]= ", w[i])  
    senao  
        escreval("k[,i,]= ", k[i])  
    fimse  
fimpara
```

IV. Incorreta.

As variáveis i, v, w, k são acessíveis em todos os módulos, mas não são locais e sim globais.