



Fundação CECIERJ - Vice Presidência de Educação Superior a Distância

Curso de Tecnologia em Sistemas de Computação
Disciplina: Projeto e Desenvolvimento de Algoritmos
AP3 2º semestre de 2006.

Nome –

Assinatura –

Observações:

1. Prova sem consulta e sem uso de máquina de calcular.
 2. Use caneta para preencher o seu nome e assinar nas folhas de questões e nas folhas de respostas.
 3. Você pode usar lápis para responder as questões.
 4. Ao final da prova devolva as folhas de questões e as de respostas.
 5. Todas as respostas devem ser transcritas nas folhas de respostas. As respostas nas folhas de questões não serão corrigidas.
-

1ª questão (valor 1.0)

O que acontecerá se o código a seguir for executado?

início

```
sum ← 0
j ← 10
i ← 0
enquanto sum > 20 faça
    sum ← sum + i + j
    i ← i + 1
    j ← j - 1
fim enquanto
imprima 'Sum = ', sum
```

fim

- A) Ele imprimirá Sum = 0 ← Correta
- B) Ele imprimirá Sum = 20
- C) Nada será impresso uma vez que existe um erro na sintaxe do PETEQS
- D) O algoritmo entrará em um laço eterno
- E) Nenhuma das respostas anteriores

2ª questão (valor 1.0)

O que será impresso pelo algoritmo a seguir? Considere a existência de uma função *tamanho(vetor)* que retorna o número de elementos em um vetor.

Observações:

- o índice do primeiro elemento de um vetor é sempre igual a um.

- o comando *imprima* não muda automaticamente de linha após a impressão

```

procedure muda(entradas: vetor[])
início
    vetor[tamanho(vetor)] ← 0
fim

início
    vetor[1] ← 1
    vetor[2] ← 2
    vetor[3] ← 3
    muda(vetor)
    para i ← 1 até tamanho(vetor) faça
        imprima vetor[i], ' '
    próximo i
fim

```

- A) 0 0 0
 B) 1 0 3
 C) 1 2 0
 D) 0 2 3
 E) 1 2 3 ← Correta

3ª questão (valor 1.0)

O que será impresso pelo algoritmo a seguir?

```

procedimento muda(entrada: n)
início
    n ← n + 1
fim

procedimento naoMuda(saída: n[])
início
    n[1] ← n[1] + 1
fim

início
    b ← 15
    muda(b)
    imprime b

    aux[1] ← b
    naoMuda(aux)
    b ← aux[1]
    imprime b
fim

```

- A) 15 15
 B) 15 16 ← Correta
 C) 16 15
 D) 16 16
 E) 16 17

O que será impresso pelo programa a seguir?

Observações:

- o comando *imprima* não muda automaticamente de linha após a impressão
- considere a existência de uma função *mudaLinha()* que pula para a linha seguinte na impressão.

```
início
  para i ← 1 até 10 faça
    para j ← 1 até 10 faça
      se j<=i então
        imprima ' * '
      fim se
    proximo i
  mudaLinha()
próximo i
fim
```

[illegible]

5ª questão (valor 1.0)

O que será impresso pelo algoritmo a seguir?

Observação:

- o comando *imprima* muda automaticamente de linha após a impressão

```
início  
  count ← 1  
  enquanto count ≤ 10 faça  
    se count mod 2 = 1 então  
      imprima '*****'  
    senão  
      imprima '+++++++'  
    fim se
```

```

    count ← count + 1
fim enquanto
fim

```

A) ← Correta	B)	C)	D)	E)
****	+++++++	***	+++++++	Nenhuma
****	****	*	+++++++	das respostas
+++++++	+++++++	***	+++++++	anteriores
****	****	*	+++++++	
+++++++	+++++++	***	+++++++	
****	****	*	+++++++	
+++++++	+++++++	***	+++++++	
****	****	*	+++++++	
+++++++	+++++++	***	+++++++	
****	****	*	+++++++	
+++++++		***		
		*		

		*		

		*		

		*		

		*		

6ª questão (valor 1.0)

Nesta questão, além dos operadores lógicos que você já conhece (E, OU e NÃO), considere o operador XOR (ou exclusivo) cuja tabela verdade é dada a seguir:

XOR ou-exclusivo

tabela verdade do ou-exclusivo:

A	B	A XOR B
falso	falso	Falso
falso	verdadeiro	verdadeiro
verdadeiro	falso	verdadeiro
verdadeiro	verdadeiro	Falso

Considere ainda os operandos:

1 equivalente a verdadeiro

0 equivalente a falso,

A expressão a seguir que, se avaliada, resultará no valor VERDADEIRO é:

A) NÃO 1

B) (((NÃO (0 E 1)) OU 0) XOR ((0 OU 1) E 1))

C) (((0 E 1) OU 0) XOR ((0 E 1) E NÃO 1))

D) (((1 E 0) OU 1) XOR ((1 E 0) E NÃO 0)) ← Correta

E) (((1 E 0) OU 1) XOR ((1 E 0) OU NÃO 0))

7ª questão (valor 1.0)

Qual das expressões abaixo produz, para quaisquer valores das variáveis lógicas A, C e D, os mesmos resultados que a expressão:

F = NÃO A E C OU A E NÃO C E NÃO D

se for assegurado que A e C não são nunca verdadeiros ao mesmo tempo?

- A) F = NÃO D
- B) F = A E C OU C E NÃO D
- C) F = C OU NÃO D
- D) F = C OU A E NÃO D ← Correta
- E) F = C E NÃO A OU D

8ª questão (valor 1.0)

O que será impresso pelo algoritmo a seguir? Considere a existência de uma função *tamanho(vetor)* que retorna o número de elementos em um vetor.

Observações:

- o índice do primeiro elemento de um vetor é sempre igual a um.
- considere que o comando *imprima* não muda automaticamente de linha após a impressão

```

procedimento altera(saída: n[])
início
    final ← tamanho(n)
    para i ← 1 até tamanho(n) faça
        temp ← vetor[i]
        vetor[i] ← vetor[final]
        vetor[final] ← temp
        final ← final - 1
    próximo i
fim

início
    vetor[1] ← 'a'
    vetor[2] ← 'b'
    vetor[3] ← 'c'
    vetor[4] ← 'd'
    vetor[5] ← 'e'
    altera(vetor)
    para i ← 1 até tamanho(vetor) faça
        imprima vetor[i], ' '
    próximo i
fim

```

- A) a b c d e ← Correta
- B) c a b d e
- C) d e c a b
- D) d e c b a
- E) e d c b a

9ª questão (valor 1.0)

O que será impresso pelo algoritmo a seguir?

Observações:

- o índice do primeiro elemento de um vetor é sempre igual a um.
- considere que o comando *imprima* não muda automaticamente de linha após a impressão

```

início
  para i ← 1 até 5 faça
    vetor[i] ← 0
  próximo i
  para i ← 1 até 5 faça
    j ← 1
    enquanto i*j ≤ 5 faça
      vetor[i*j] ← vetor[i*j] + 1
      j ← j + 1
    fim enquanto
  próximo i
  para i ← 1 até 5 faça
    imprima vetor[i], ' ', '
  próximo i
fim

```

- A) 1, 2, 2, 3, 2, ← Correta
 B) 1, 2, 3, 4, 5,
 C) 1, 3, 5, 7, 9,
 D) 1, 2, 4, 8, 16,
 E) 1, 5, 10, 15, 20,

10ª questão (valor 1.0)

O que será impresso pelo algoritmo a seguir?

Observações:

- o índice do primeiro elemento de um vetor é sempre igual a um.
- considere que o comando *imprima* não muda automaticamente de linha após a impressão

```

início
  vetor[1] ← 4.0
  vetor[2] ← 3.0
  vetor[3] ← 7.0
  vetor[4] ← 2.0
  soma1 ← 0.0
  soma2 ← 0.0
  para i ← 1 até 4 faça
    soma1 ← soma1 + vetor[i]
    soma2 ← (soma2 + vetor[i])/4
  próximo i
  soma1 ← soma1 / 4
  imprima soma1, ' ', ' ', soma2
fim

```

- A) 0.25, 0.0625
 B) 1.0, 4.0
 C) 1.75, 0.125
 D) 4.0, 1.0 ← Correta
 E) 4.0, 4.0