



Fundação CECIERJ - Vice Presidência de Educação Superior a Distância

Curso de Tecnologia em Sistemas de Computação
Disciplina: Projeto e Desenvolvimento de Algoritmos
AP2 2º semestre de 2010.

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 no local apropriado, no cartão de respostas a seguir.

Questão					
1	A	B	<input checked="" type="checkbox"/>	D	E
2	A	B	C	<input checked="" type="checkbox"/>	E
3	A	<input checked="" type="checkbox"/>	C	D	E
4	<input checked="" type="checkbox"/>	B	C	D	E
5	A	<input checked="" type="checkbox"/>	C	D	E
6	A	B	C	<input checked="" type="checkbox"/>	E
7	A	B	<input checked="" type="checkbox"/>	D	E
8	A	B	<input checked="" type="checkbox"/>	D	E
9	<input checked="" type="checkbox"/>	B	C	D	E
10	A	<input checked="" type="checkbox"/>	C	D	E

1ª questão (valor 1.0)

Na análise do algoritmo a seguir, considere a existência da função **tamanho(entradas: n[])** que retorna o número de elementos no array **n**.

```
procedimento operacoesAritmeticas(entradas: números[])  
início
```

```
    soma ← 0  
    para i ← 1 até tamanho(numeros) faça  
        soma ← soma + numeros[i]  
    próximo i  
    imprima soma / (tamanho(numeros) + 0.0)  
    imprima soma  
    para i ← 1 até tamanho(numeros) faça  
        imprima soma MOD numeros[i]  
    próximo i
```

```
fim
```

```
início
```

```
    para i ← 1 até 5 faça  
        imprima "num", i, "? "  
        leia num[i]  
    próximo i  
    operacoesAritmeticas(num)
```

```
fim
```

Se os valores fornecidos pelo usuário forem 12, 13, 14, 15 e 16, nessa ordem, a saída do programa será: (as respostas foram formatadas em colunas apenas para melhor visualização)

- A) 14.0 70 5 5 5 4 4
B) 35 70 11 9 7 5 3
C) 14.0 70 10 5 0 10 6
D) 0 70 12 13 14 15 16
E) Nenhuma das respostas anteriores

2ª questão (valor 1.0)

Observe o algoritmo a seguir:

```
procedimento main()
```

```
início
```

```
    gJ ← 15000  
    gF ← 23000  
    gM ← 17000  
    gT ← gJ / (gF + gM)
```

```
fim
```

```
início
```

```
    gT ← 1  
    main()  
    imprima gT
```

```
fim
```

A saída impressa pelo algoritmo será:

- A) 0
- B) 0.1
- C) 0.375
- D) 1
- E) Nenhuma das respostas anteriores

3ª questão (valor 1.0)

Observe o algoritmo a seguir:

```
procedimento main3()
início
    DIM ← 5
    media ← 0
    para i ← 1 até DIM faça
        imprima "n", i, "? "
        leia n[i]
    próximo i
    para i ← 1 até DIM faça
        media ← media + n[i]/(DIM + 0.0)
    próximo i
    se media > 9.0 então
        imprima "A1"
    senão
        se media >= 5.0 então
            imprima "A2"
        senão
            imprima "A3"
    fim se
fim se
fim

início
    main3()
fim
```

Se os valores fornecidos pelo usuário ao algoritmo forem 8.0, 8.0, 8.0, 10.0 e 9.0, o valor impresso pelo algoritmo será:

- A) A1
- B) A2
- C) A3
- D) A1 A2
- E) Nenhuma das respostas anteriores

4ª questão (valor 1.0)

Na análise do algoritmo a seguir, considere a existência da função `tamanho(entradas: n[])` que retorna o número de elementos no array `n`.

```

procedimento m1(entradas: array[])
início
    ma ← array[1]
    me ← array[1]
    para i ← 2 até tamanho(array) faça
        se array[i] > ma então
            ma ← array[i]
        fim se
        se array[i] < me então
            me ← array[i]
        fim se
    próximo i
    imprima ma
    imprima me
fim

início
    DIM ← 10
    para i ← 1 até DIM faça
        imprima "array[" , i, "]"? "
        leia array[i]
    próximo i
    m1(array)
fim

```

Se os valores fornecidos pelo usuário forem 12, 45, 67, 89, 9, 65, 43, 21, 10 e -89, nessa ordem, a saída do programa será: (as respostas foram formatadas em colunas apenas para melhor visualização)

- A) 89 -89
- B) 89 9
- C) -89 -89
- D) 89 89
- E) Nenhuma das respostas anteriores

5ª questão (valor 1.0)

Na análise do algoritmo a seguir, considere a existência da função **tamanho(entradas: n[])** que retorna o número de elementos no array **n**.

```

função am(entradas: ini
           saídas: array[])
início
    m ← array[ini]
    resultado ← ini
    para i ← (ini + 1) até tamanho(array) faça
        se array[i] < m então
            m ← array[i]
            resultado ← i
        fim se
    próximo i
fim

```

```

procedimento te(entradas: i, j
                saídas: array[])
início
    aux ← array[i]
    array[i] ← array[j]
    array[j] ← aux
fim

procedimento ord(saídas: array[])
início
    para esq ← 1 até tamanho(array) faça
        dir ← am(esq, array)
        te(esq, dir, array)
    próximo esq
fim

procedimento org(entradas: array
                 saídas: saída)
início
    j ← 1
    k ← tamanho(array)
    ord(array)
    para i ← 1 até tamanho(array) faça
        se array[i] MOD 2 = 0 então
            saída[j] ← array[i]
            j ← j + 1
        senão
            saída[k] ← array[i]
            k ← k - 1
        fim se
    próximo i
fim

início
    DIM ← 10
    para i ← 1 até DIM faça
        leia array[i]
    próximo i
    org(array, array)
    para i ← 1 até DIM faça
        imprima array[i]
    próximo i
fim

```

Se os valores fornecidos pelo usuário forem 12, 45, 67, 89, 9, 66, 43, 22, 10 e 8, nessa ordem, a saída do programa será: (as respostas foram formatadas em colunas apenas para melhor visualização)

- A) 8 9 10 12 22 43 45 66 67 89
- B) 8 10 12 22 66 89 67 45 43 9
- C) 89 67 66 45 43 22 12 10 9 8
- D) 66 22 12 10 8 9 43 45 67 89
- E) Nenhuma das respostas anteriores

6ª questão (valor 1.0)

Observe o algoritmo a seguir.

```
função f ( entradas: n)
início
    m ← 1
    para i ← 1 até n faça
        m ← m * i
    próximo i
    resultado ← m
fim

início
    leia n
    imprima f(n)
fim
```

Se a entrada fornecida pelo usuário for 5
o valor impresso pelo algoritmo será:

- A) 60
- B) 1
- C) 4
- D) 120
- E) Nenhuma das respostas anteriores

7ª questão (valor 1.0)

Observe o algoritmo a seguir:

```
função p ( entradas: n)
início
    r ← 0.0
    s ← 1
    d ← 1.0
    i ← 0
    enquanto i < n faça
        r ← r + s * (1 / d)
        s ← -1 * s
        d ← d + 2
        i ← i + 1
    fim enquanto
    resultado ← r
fim

início
    imprima 4 * p(4)
fim
```

A saída impressa pelo algoritmo está no intervalo:

- A) maior que 0 e menor ou igual a 1
- B) maior que 1 e menor ou igual a 2
- C) maior que 2 e menor ou igual a 3
- D) maior que 3 e menor ou igual a 4
- E) Nenhuma das Respostas Anteriores

8ª questão (valor 1.0)

Observe o algoritmo a seguir:

```
função p ( entradas: v[], n, x)
início
    y ← 0.0
    i ← n
    enquanto (i > 0) faça
        y ← y * x + v[i]
        i ← i - 1
    fim enquanto
    resultado ← y
fim

início
    leia x
    leia n
    para i ← 1 até n faça
        leia c[i]
    próximo i
    imprima p(c, n, x)
fim
```

Se os dados de entrada fornecidos pelo usuário forem 2, 3, 3, 2 e 1, a saída impressa pelo algoritmo será:

- A) 1
- B) 7
- C) 11
- D) 5
- E) Nenhuma das Respostas Anteriores

9ª questão (valor 1.0)

Observe o algoritmo a seguir:

```
função m ( entradas: v[], t)
início
    mm ← v[1]
    para i ← 2 até t
        se mm < v[i] então
            mm ← v[i]
        fim se
    próximo i
    resultado ← mm
fim
```

```

início
    para i ← 1 até 5 faça
        leia c[i]
    próximo i
    imprima m(v, 5)
fim

```

Se os dados de entrada fornecidos pelo usuário forem 1, 2, 3, 4 e 5, a saída impressa pelo algoritmo será:

- A) 5
- B) 4
- C) 3
- D) 2
- E) Nenhuma das respostas anteriores

10ª questão (valor 1.0)

Considere que, em PETEQS, existe uma tabela do tipo ASCII que atribui um número inteiro para cada caractere. Considere ainda que estão disponíveis as seguintes funções:

ordem(entradas: car) { retorna a posição do caractere car na tabela de caracteres do computador }

caractere(entradas: num) { retorna o caractere cuja posição na tabela de caracteres do computador seja num }

LeCadeia(saídas: frase) { lê um conjunto de caracteres do teclado e os armazena no vetor frase }

ImprimeCadeia(entradas: frase) { imprime o conjunto de caracteres do vetor frase }

CompCadeia(entradas: frase) { retorna quantos caracteres estão armazenados no vetor frase }

Os caracteres alfabéticos ocupam posições contíguas na tabela, isto é,

ordem('B') - ordem('A') = 1

e portanto,

caractere (ordem('A') + 1) = 'B'

Usando estas funções, um aluno de PDA escreveu o seguinte algoritmo:

```

início
    LeCadeia(frase)
    c ← 0
    para i ← 1 até CompCadeia(frase) faça
        se (ord(frase[i]) >= ord('A')) E (ord(frase[i]) <= ord('U')) então
            c ← c + 1
        fim se
    próximo i
    imprima c
fim

```


Marque a opção que mostra o que será impresso pelo algoritmo caso seja digitado o seguinte conjunto de caracteres:

TERCEIRA

- A) 6
- B) 8
- C) 4
- D) 2
- E) Nenhuma das respostas anteriores