



Fundação CECIERJ - Vice Presidência de Educação Superior a Distância

**Curso de Tecnologia em Sistemas de Computação**  
**Disciplina: Projeto e Desenvolvimento de Algoritmos**  
**AP2 1º semestre de 2015**

Nome –

Assinatura –

---

Observações:

- A) Prova sem consulta e sem uso de máquina de calcular.
- B) Use caneta para preencher o seu nome e assinar nas folhas de questões e nas folhas de respostas.
- C) Você pode usar lápis para responder as questões.
- D) Ao final da prova devolva as folhas de questões e as de respostas.
- E) **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	<input checked="" type="checkbox"/>	B	C	D	E
3	A	<input checked="" type="checkbox"/>	C	D	E
4	A	<input checked="" type="checkbox"/>	C	D	E
5	A	<input checked="" type="checkbox"/>	C	D	E
6	<input checked="" type="checkbox"/>	B	C	D	E
7	A	<input checked="" type="checkbox"/>	C	D	E
8	A	B	<input checked="" type="checkbox"/>	D	E
9	<input checked="" type="checkbox"/>	B	C	D	E
10	A	B	C	<input checked="" type="checkbox"/>	E

Para a análise dos algoritmos nesta prova, considere que o comando **leia** é capaz de ler uma cadeia de caracteres de uma vez. Por exemplo, o comando **leia nome** lê todos os caracteres digitados pelo usuário até que ele digite a tecla **enter** e os armazena em um vetor chamado **nome**.

Considere também a existência das funções **charAt()** e **tamanho()**, cuja documentação é mostrada a seguir:

**função charAt(entradas: str, pos)**

Retorna uma string contendo o caractere na posição **pos** da cadeia de caracteres **str** passada como parâmetro.

Exemplo:

```
imprima charAt('CEDERJ', 3)      // imprimiria 'D'
```

**função tamanho(entradas: str)**

Retorna o número de caracteres na string **str** passada como parâmetro.

Exemplos:

```
imprima tamanho('Dilma')      # imprimiria 5
```

## 1ª questão (valor 1.0)

O maior valor inteiro sem sinal que pode ser armazenado em um byte é:

- A) 127
- B) 128
- C) 255
- D) 256
- E) Nenhuma das respostas anteriores

## 2ª questão (valor 1.0)

Para essa questão, considere que o seguinte vetor de strings foi definido:

```
nomes[1] ← "Lucas"  
nomes[2] ← "Alexandre"  
nomes[3] ← "Alberto"  
nomes[4] ← "Pedro"  
nomes[5] ← "Ana"
```

O código correto para imprimir todas as strings começadas pela letra 'A' maiúscula é:

A)

```
para i ← 1 até 5 faça  
    se charAt(nomes[i], 1) = 'A' então  
        imprima nomes[i]  
    fim se  
próximo i
```

B)

```
i ← 1  
enquanto charAt(nomes[i], 1) = 'A' faça  
    imprima nomes[i]  
    i ← i + 1  
fim enquanto
```

C)

```
para i ← 1 até 5 faça  
    para j ← 1 até tamanho(nomes[i]) faça  
        se (charAt(nomes[i], j) <> 'A') OU (j <> 1) então  
            imprima nomes[i]  
        fim se  
    próximo j  
próximo i
```

D)

```
para i ← 1 até tamanho(nomes[i]) faça  
    se charAt(nomes[i], 1) = 'A' então  
        imprima nomes[i]  
    fim se  
próximo i
```

E) Nenhuma das respostas anteriores

## 3ª questão (valor 1.0)

Observe o algoritmo a seguir:

```

função proc(entradas: a, b)
início
    resultado ← a
    se test(a) então
        resultado ← b
    fim se
fim

```

Qual das opções a seguir mostra uma função que não é equivalente à função **proc()** original? Para responder a essa questão, use a seguinte definição de equivalência: “se uma função é equivalente à outra, o código que usa as funções não deveria perceber a diferença entre chamar uma ou outra”. Observe que não é preciso conhecer a função **test()** para responder à pergunta.

A)

```

função proc(entradas: x, y)
início
    se test(x) então
        resultado ← y
    senão
        resultado ← x
    fim se
fim

```

B)

```

função proc(entradas: a, b)
início
    se NÃO test(b) então
        resultado ← a
    senão
        resultado ← b
    fim se
fim

```

C)

```

função proc(entradas: a, b)
início
    resultado ← b
    se NÃO test(a) então
        resultado ← a
    fim se
fim

```

D)

```

função proc(entradas: a, b)
início
    resultado ← a
    se NÃO test(a) então
        b ← 'cederj'
    senão
        resultado ← b
    fim se
fim

```

E) Nenhuma das respostas anteriores

## 4ª questão (valor 1.0)

O que será impresso pelo algoritmo a seguir?

```

função maior2(entradas: a, b)
início
    se a > b então
        resultado ← a
    senão
        resultado ← b
    fim se
fim

função maior3(entradas: a, b, c)
início
    resultado ← maior2(a, maior2(b, c))
fim

função misterio(entradas: a, b, c)
início
    se a = maior3(a, b, c) então
        resultado ← maior2(b, c)
    senão
        se b = maior3(a, b, c) então
            resultado ← maior2(a, c)
        senão
            resultado ← maior2(a, b)
        fim se
    fim se
fim

início
    imprima misterio(9,3,6)
fim

```

- A) 3
- B) 6
- C) 9
- D) 18
- E) Nenhuma das respostas anteriores

## 5ª questão (valor 1.0)

O que será impresso pelo algoritmo a seguir?

```

procedimento notas(entradas: valor)
início
    cinco ← valor / 5
    dois ← (valor MOD 5) / 2
    um ← dois MOD 2
    imprima cinco, " ", dois, " ", um
fim

início
    notas(29)
fim

```

- A) 5 2 1
- B) 5 2 0
- C) 1 1 1
- D) 1 0 0
- E) Nenhuma das respostas anteriores

## 6ª questão (valor 1.0)

Observe o algoritmo a seguir.

```
procedimento le(entradas: tam, saídas: v[])
início
    para i ← 1 até tam faça
        leia v[i]
    próximo i
fim

função m (entradas: tam, v[])
início
    s ← 0
    para i ← 1 até tam faça
        s ← s + v[i]
    próximo i
    s ← s / tam
    resultado ← s
fim

função va (entradas: tam, v[], m)
início
    var ← 0.0
    para i ← 1 até tam faça
        var ← var + (v[i] - m) * (v[i] - m)
    próximo i
    resultado ← var/tam
fim

início
    le (4, v)
    med ← m(4, v)
    imprima va(4, v, med)
fim
```

Se os dados de entrada fornecidos pelo usuário forem 5.0, 8.0, 4.0 e 3.0, nesta ordem, a saída impressa pelo algoritmo será:

- A) 3.5
- B) 4.5
- C) 5.5
- D) 6.5
- E) Nenhuma das respostas anteriores

## 7ª questão (valor 1.0)

Observe o algoritmo a seguir:

```
início
    v ← 0
    para i ← 1 até 3 faça
        para j ← i até 3 faça
            v ← v + i * j
        próximo j
    próximo i
    imprima v
fim
```

A saída impressa pelo algoritmo será:

- A) 15
- B) 25
- C) 35
- D) 45
- E) Nenhuma das respostas anteriores

## 8ª questão (valor 1.0)

Observe o algoritmo a seguir:

```
início
    str ← 'ABCDEF'
    num ← 'ABA'
    val ← 0
    para i ← 1 até tamanho(num) faça
        para j ← 1 até tamanho(str) faça
            se charAt(num,i) = charAt(str, j) então
                val ← val + 10 + j - 1
            fim se
        próximo j
    próximo i
    imprima val
fim
```

A saída impressa pelo algoritmo será:

- A) 11
- B) 21
- C) 31
- D) 41
- E) Nenhuma das respostas anteriores

## 9ª questão (valor 1.0)

Observe o algoritmo a seguir:

```

início
  pts ← 0
  para i ← 1 até 4 faça
    camp[i] ← 0
  próximo i
  gc ← 0
  gv ← 0
  leia tc, gc, gv, tv
  enquanto (gc >= 0) E (gv >= 0) faça
    se gc > gv então
      camp[tc] ← camp[tc] + 3
    senão
      se gc = gv então
        camp[tc] ← camp[tc] + 1
        camp[tv] ← camp[tv] + 1
      senão
        camp[tv] ← camp[tv] + 3
      fim se
    fim se
  leia tc, gc, gv, tv
fim enquanto
para i ← 1 até 4 faça
  imprima camp[i]
próximo i
fim

```

Se os dados de entrada fornecidos pelo usuário forem

```

1 0 0 2
1 2 0 3
1 2 2 4
2 1 1 3
2 2 1 4
3 1 3 4
-1 -1 -1 -1

```

A saída impressa pelo algoritmo será:

- A) 5 5 1 4
- B) 1 2 3 4
- C) 5 4 1 3
- D) 5 1 1 1
- E) Nenhuma das respostas anteriores



## 10ª questão (valor 1.0)

Observe o algoritmo a seguir:

```
procedimento p(entradas: x, y, saídas: z)
    x ← x + 1
    z ← x + y
fim
```

```
função f(entradas: x, y, z)
    x ← x + 1
    z ← x + y
    resultado ← z
fim
```

```
início
    a ← 10
    b ← 15
    c ← 0
    p(a, b, c)
    imprima a, b, c
    c ← f(a, b, c)
    imprima a, b, c
fim
```

A saída impressa pelo algoritmo será:

- A) 10 15 26  
10 15 15
- B) 11 15 26  
10 15 25
- C) 11 15 25  
11 15 26
- D) 10 15 26  
10 15 26
- E) Nenhuma das respostas anteriores