



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 2014**

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

Para a análise dos algoritmos nessa prova, considere a existência das funções `abs()`, `find()`, cuja documentação é mostrada a seguir:

**função `abs(entradas: num)`**

Retorna o valor absoluto do número num.

Exemplo:

```
imprima abs(-10)           # imprimiria 10
imprima abs(10)            # imprimiria 10
```

**função `find(entradas: str1, str2)`**

Retorna o índice em str1 onde foi encontrada a string str2. Retorna -1 se a string str2 não for encontrada.

Exemplo:

```
imprima find('Novo Teste', 'ovo')    # imprimiria 2
imprima find('Novo Teste', 'Cas')    # imprimiria -1
imprima find('Novo Teste', 'Tes')    # imprimiria 6
imprima find('Novo Teste', 'a')      # imprimiria -1
```

Considere ainda a existência de uma forma alternativa de inicializar variáveis indexadas em PETEQS:

```
V ← [1, 2, 3]
```

que é equivalente a:

```
V[1] ← 1
```

```
V[2] ← 2
```

```
V[3] ← 3
```

## 1ª questão (valor 1.0)

Uma forma de calcular o risco de um ativo financeiro é dividir a média dos retornos diários pelo desvio médio no período. O algoritmo a seguir calcula o risco para um dado ativo.

**início**

```
retornos ← [1.0, -0.5, -2, 1.5, 2.5]
media ← 0
para i ← 1 até 5 faça
    media ← media + retornos[i]/5.0
próximo i
desvio ← 0
para i ← 1 até 5 faça
    desvio ← desvio + abs(retornos[i] - media)/5.0
próximo i
sharpe ← media/desvio
imprima sharpe
```

**fim**

O valor impresso pelo algoritmo é:

- A) entre 0.25 e 0.27
- B) entre 0.30 e 0.32
- C) entre 0.35 e 0.37
- D) entre 0.40 e 0.42
- E) entre 0.45 e 0.47

## 2ª questão (valor 1.0)

Uma maneira de tentar obter lucros no mercado financeiro é investigar o comportamento de um ativo nos dias seguintes à ocorrência de um evento. Um evento é definido como uma queda do preço de fechamento do ativo em um dia de alta do mercado. O algoritmo a seguir conta o número de eventos em um dado período.

**início**

```
IBX ← [1.0, 1.0, 1.02, 1.0, 0.99, 1.0, 1.05, 1.0, 0.95, 1]
close ← [1.0, 1.0, 1.02, 1.0, 0.95, 1.0, 0.95, 1.0, 1.05, 1]
eventos ← 0
para i ← 2 até 10 faça
    se (close[i]/close[i-1] - 1 < -0.03) E
        (IBX[i]/IBX[i-1] - 1 > 0.02) então
        eventos ← eventos + 1
    fim se
próximo i
imprima eventos
```

**fim**

O valor impresso pelo algoritmo é:

- A) 1
- B) 2
- C) 3
- D) 4
- E) 5

### 3ª questão (valor 1.0)

O algoritmo a seguir procura a composição de uma carteira formada por dois ativos de modo que o retorno mensal seja zerado.

```
início
    retorno ← [-1.0, 0.25]
    pa ← 0
    enquanto pa ≤ 1.0 faça
        pb ← 1.0 - pa
        se (pa*retorno[1] + pb*retorno[2]) = 0 então
            imprima pa, ' ', pb
        fim se
        pa ← pa + 0.1
    fim enquanto
fim
```

O valor impresso pelo algoritmo é:

- A) 0.2    0.8
- B) 0.4    0.6
- C) 0.6    0.4
- D) 0.8    0.2
- E) 1.0    0.0

### 4ª questão (valor 1.0)

O que será impresso pelo programa a seguir?

```
início
    DIM ← 20
    seq ← [4, 56, 3, 8, 0, 0, 0, 0, 4, 7, 9, 13, 25, 3, 0, 7, 0, 1, 2, 3]
    grupos ← 0
    para i ← 2 até DIM faça
        se (seq[i] = 0) E (seq[i-1] <> 0) então
            grupos ← grupos + 1
        fim se
    próximo i
    se seq[DIM] <> 0 então
        grupos ← grupos + 1
    fim se
    imprima grupos
fim
```

- A) 0
- B) 1
- C) 2
- D) 3
- E) 4

### 5ª questão (valor 1.0)

O que será impresso pelo algoritmo a seguir?

```

início
    j ← 1
    enquanto j < 10 faça
        j ← j + j MOD 3
    fim enquanto
    imprima j
fim

```

- A) 8
- B) 9
- C) 10
- D) 11
- E) 12

## 6ª questão (valor 1.0)

O que será impresso pelo algoritmo a seguir, se o valor fornecido for 10?  
 Considere que o comando **imprima** não muda de linha ao imprimir.

```

procedimento tr1 (entradas: t1)
início
    t1 ← t1 + 10
fim

```

```

procedimento tr2 (saídas t1)
início
    t1 ← t1 - 10
fim

```

```

início
    leia a
    tr1 (a)
    imprima a, ' '
    tr2 (a)
    imprima a
fim

```

- A) 20    10
- B) 10    0
- C) 10    10
- D) 0    10
- E) Nenhuma das respostas anteriores

## 7ª questão (valor 1.0)

O que será impresso pelo algoritmo a seguir?

```

função misterio(entradas: frase[])
    cont ← 0
    teste ← 'aeiou'
    i ← 1
    enquanto (i <= 5) faça
        se (find(frase, teste[i]) < 0) então
            cont ← cont + 1
        fim se
        i ← i + 1
    fim enquanto
    resultado ← cont
fim

```

```

início
    texto ← 'frase'
    imprima 'Nao tem ', misterio(texto)
fim

```

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

## 8ª questão (valor 1.0)

O que será impresso pelo algoritmo a seguir, se os valores fornecidos forem 2 4 0 6?  
 Considere que o comando **imprima** não muda de linha ao imprimir.

```

procedimento troca (saídas: a, b)
início
    se (a > b) então
        t ← a
        a ← b
        b ← t
    fim se
fim

```

```

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

```

```

procedimento imprime(entradas: t, saídas: v)
início
    para i ← 1 até t faça
        imprima v[i], ' '
    próximo i
fim

```

```

procedimento naosei(entradas: t, saídas: v)
início
    ordem ← 0
    enquanto (ordem = 0) faça
        ordem ← 1
        para i ← 1 até t-1 faça
            se v[i] > v[i+1] então
                troca(v[i], v[i+1])
                ordem ← 0
            fim se
        fim para
    fim enquanto
fim

```

```

início
    le(4, v)
    naosei(4, v)
    imprime(4, v)
fim

```

- A) 2 4 0 6
- B) 6 4 2 0
- C) 4 2 6 0
- D) 0 2 4 6
- E) Nenhuma das respostas anteriores

## 9ª questão (valor 1.0)

Considere o algoritmo a seguir:

```

início
    tp ← 0
    im ← 0
    leia n
    enquanto n <> 0 faça
        to ← to + 1
        se (n mod 2) = 1 então
            im ← im + 1
        fim se
        leia n
    fim enquanto
    imprima to, ' - ', im
fim

```

Se os valores fornecidos pelo usuário ao algoritmo forem 7, 6, 5, 4, 3, 2, 1 e 0, respectivamente, a saída impressa será:

- A) 8 - 4
- B) 8 - 3
- C) 7 - 3
- D) 7 - 4
- E) Nenhuma das respostas anteriores

## 10ª questão (valor 1.0)

Considere o algoritmo a seguir:

```
função m (entradas: v[], t)
início
    c ← 0.0
    para i ← 1 até t faça
        c ← c + v[i]
    próximo i
    resultado ← c / t
fim
```

```
função q(entradas: v[], t)
início
    ma ← 0
    c ← m(v, t)
    para i ← 1 até t faça
        se (v[i] > c) então
            ma ← ma + 1
        fim se
    próximo i
    resultado ← ma
fim
```

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

Se os valores fornecidos ao algoritmo forem 2, 4, 6, 8 e 10, respectivamente, a saída impressa será o número:

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