

**Curso de Tecnologia em Sistemas de Computação**  
**Disciplina: Projeto e Desenvolvimento de Algoritmos**  
**AP2 2º semestre de 2012**

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

Para a análise dos algoritmos nessa prova, considere a existência das funções `abs()`, `concat()`, `tamanho()`, `charAt()`, `split()` e `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 `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 `concat(entradas: str1, str2)`**

Retorna uma cadeia de caracteres formada pela concatenação de `str1` e `str2`.

Exemplo:

```
imprima concat('Alo ', 'mundo!') // imprimiria 'Alo mundo!'
```

**função `split(entradas: str)`**

Retorna um vetor de strings contendo as palavras na string de entrada `str`. As palavras são delimitadas por espaços em branco. A última posição do vetor de saída é preenchida com a string vazia ('').

Exemplo:

```
imprima split('carro velho')    // ['carro', 'velho', '']
```

**função `tamanho(entradas: str)`**

Retorna o número de caracteres na cadeia de caracteres passada como parâmetro.

Exemplo:

```
imprima tamanho('CEDERJ')       // imprimiria 6
```

**função `find(entradas: str, ch)`**

Retorna o índice em `str` onde foi encontrado o caractere `ch`. Retorna -1 se o caractere `ch` não for encontrado.

Exemplo:

```
imprima find('Dilma Roussef', 'l') // imprimiria 3
imprima find('Dilma Roussef', 'z') // imprimiria -1
```

## 1ª questão (valor 1.0)

O que será impresso pelo algoritmo a seguir?

```
função decodifica(entradas: mensagem)
```

```
início
```

```
    resultado ← ''
```

```
    palavras ← split(mensagem)
```

```
    i ← 1
```

```
    enquanto palavras[i] <> '' faça
```

```
        resultado ← concat(resultado, charAt(palavras[i], 1))
```

```
        i ← i + 1
```

```
    fim enquanto
```

```
fim
```

```
início
```

```
    imprima decodifica('leitura intelectual volume reduzido obvio')
```

```
fim
```

A) leitura

B) livro

C) intelectual

D) obvio

E) Nenhuma das respostas anteriores

## 2ª questão (valor 1.0)

O que será impresso pelo algoritmo a seguir?

```
funcao misterio(entradas: lista)
```

```
início
```

```
    aux ← 'ijklmno'
```

```
    resultado ← ''
```

```
    i ← tamanho(lista)
```

```
    enquanto i > 2 faça
```

```
        se i < (tamanho(lista)/2 + i MOD 5) então
```

```
            resultado ← concat(resultado, charAt(aux, (i - 2)*(i - 2)))
```

```
        senão
```

```
            resultado ← concat(resultado, charAt(lista, i))
```

```
        fim se
```

```
        i ← i - 1
```

```
    fim enquanto
```

```
fim
```

```
início
```

```
    letras ← 'abcdef'
```

```
    imprima misterio(letras)
```

```
fim
```

A) fedc

B) fedcba

C) feli

D) fedlkj

E) Nenhuma das respostas anteriores

### 3ª questão (valor 1.0)

O que será impresso pelo algoritmo a seguir?

```
função quemSabe(entradas: primeiro, segundo)
início
    resultado ← verdadeiro
    se tamanho(primeiro) <> tamanho(segundo) então
        resultado ← falso
    senão
        para i ← 1 até tamanho(primeiro) faça
            para j ← i + 1 até tamanho(primeiro) faça
                se charAt(primeiro, i) = charAt(primeiro, j) E
                   charAt(segundo, j) <> charAt(segundo, i) então
                    resultado ← falso
            fim se
        próximo j
    próximo i
    fim se
fim

procedimento misterio(entrada: padrao, palavras[])
início
    i ← 1
    enquanto palavras[i] <> '' faça
        palavra ← palavras[i]
        se quemSabe(palavra, padrao) E quemSabe(padrao, palavra) então
            imprima palavra
        fim se
        i ← i + 1
    fim enquanto
fim

início
    palavras[1] ← 'ABACATE'
    palavras[2] ← 'LARANJA'
    palavras[3] ← 'ABACAXI'
    palavras[4] ← ''
    misterio('XCXDXUF', palavras)
fim
```

- A) ABACATE LARANJA ABACAXI
- B) ABACATE
- C) LARANJA ABACAXI
- D) ABACATE ABACAXI
- E) Nenhuma das respostas anteriores

### 4ª questão (valor 1.0)

O que será impresso pelo algoritmo a seguir?

```

função misterio(entradas: s1, s2)
início
    resultado ← ''
    para i ← 1 até tamanho(s1) faça
        ch ← charAt(s1, i)
        se find(s2, ch) <> -1 então
            resultado ← concat(resultado, ch)
        fim se
    próximo i
fim

início
    imprima misterio('abracadabra', 'ra')
fim

```

- A) ar
- B) ra
- C) aracadara
- D) araaaara
- E) Nenhuma das respostas anteriores

## 5ª questão (valor 1.0)

Assuma que a variável **instrutores** contém o valor inteiro 2. A expressão booleana a seguir que resulta no valor **verdadeiro** é:

- A) (**instrutores** < 2) OU (**instrutores** > 4)
- B) **NÃO** (**instrutores** < 3)
- C) **NÃO**(**NÃO**(**instrutores** >= -3))
- D) **NÃO**(**instrutores** \* 2 > 1)
- E) Nenhuma das respostas anteriores

## 6ª questão (valor 1.0)

Se o valor fornecido ao algoritmo a seguir for 30, a saída impressa será:

```

início
  leia x
  c1[1] ← 400
  c1[2] ← 150
  c1[3] ← 210
  c1[4] ← 300
  c2[1] ← 180
  c2[2] ← 310
  c2[3] ← 200
  c2[4] ← 145
  m1 ← abs(c1[1] - x)
  m2 ← abs(c2[1] - x)
  para i ← 2 até 4 faça
    t1 ← abs(c1[i] - x)
    t2 ← abs(c2[i] - x)
    se t1 < m1 então
      m1 ← t1
    fim se
    se t2 < m2 então
      m2 ← t2
    fim se
  próximo i
  se m2 > m1 então
    imprima m2, m1
  senão
    imprima m1, m2
  fim se
fim

```

- A) 115 120
- B) 120 115
- C) 400 300
- D) 150 145
- E) Nenhuma das respostas anteriores

## 7ª questão (valor 1.0)

O algoritmo a seguir calcula o valor do polinômio  $4x^3 + x^2 + x + 5$

```

início
  leia x
  c[1] ← 4
  c[2] ← +1
  c[3] ← -1
  c[4] ← 5
  p ← c[1]
  para i ← 2 até 4 faça
    imprima p
    p ← p * x + c[i]
  próximo i
  imprima p
fim

```

Considere que, neste algoritmo, o comando **imprima** não muda de linha após a impressão. Se o valor fornecido ao algoritmo for 2, os valores impressos serão:

- A) 4 9 17 39
- B) 4 10 25 39
- C) 4 5 4 9
- D) 4 4 -4 20
- E) Nenhuma das respostas anteriores

## 8ª questão (valor 1.0)

Se o valor fornecido ao algoritmo a seguir for o número inteiro 57, o valor impresso será:

```
início
    vetor[1] ← 2      vetor[2] ← 8
    vetor[3] ← 12     vetor[4] ← 25
    vetor[5] ← 34     vetor[6] ← 45
    vetor[7] ← 56     vetor[8] ← 57
    vetor[9] ← 78     vetor[10] ← 84
    leia chave
    i ← 1
    f ← 10
    naoAchou ← verdadeiro
    enquanto (i <= f) E (naoAchou = verdadeiro) faça
        m ← (i + f) / 2
        se (vetor[m] = chave) então
            imprima m
            naoAchou ← falso
        senão
            se (vetor[m] > chave) então
                f ← m - 1
            senão
                i ← m + 1
            fim se
        fim se
    fim enquanto
    se naoAchou = verdadeiro então
        imprima -1
    fim se
fim
```

- A) 57
- B) 28
- C) 8
- D) -1
- E) Nenhuma das respostas anteriores

## 9ª questão (valor 1.0)

Observe o algoritmo a seguir:

```

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

procedimento misterio(entradas: tam, saídas: v[])
início
    para i ← 1 até tam-1 faça
        se (v[i] < v[i+1]) então
            tmp ← v[i]
            v[i] ← v[i+1]
            v[i+1] ← tmp
        fim se
    próximo i
fim

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

início
    tam ← 5
    leVetor(tam, v)
    misterio(tam, v)
    impVetor(tam, v)
fim

```

Considere que, neste algoritmo, o comando **imprima** não muda de linha após a impressão. Caso os valores fornecidos ao algoritmo sejam **1, 2, 3, 4 e 5**, nesta ordem, quais seriam os valores impressos?

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

## 10ª questão (valor 1.0)

O que será impresso pelo algoritmo a seguir?

```

procedimento f1(entradas: a)
início
    a ← a + 1
fim

procedimento f2(saídas: a)
início
    a ← a + 1
fim

função f3(entradas: a)

```



```
início  
    resultado ← a + 1  
fim
```

```
início  
    a ← 10  
    f1(a)  
    imprima a  
    a ← 10  
    f2(a)  
    imprima a  
    a ← 10  
    a = f3(a)  
    imprima a  
fim
```

- A) 10 10 10
- B) 11 11 11
- C) 10 11 11
- D) 10 11 12
- E) Nenhuma das respostas anteriores