

Curso de Tecnologia em Sistemas de Computação
Disciplina: Projeto e Desenvolvimento de Algoritmos
AP2 2º semestre de 2013

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

Para a análise dos algoritmos nessa prova, considere a existência das funções `charAt()`, `ordem()`, `find()` 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 ordem(entradas: car)

Retorna o valor ASCII do caracter **car**

Exemplo:

```
imprima ordem('A') // imprimiria 65
```

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('Dilma Roussef', 'a R') # imprimiria 5
imprima find('Dilma Roussef', 'ilma') # imprimiria 2
imprima find('Dilma Roussef', 'ousef') # imprimiria -1
```

função tamanho(entradas: vetor[])

Retorna o número de elementos no vetor passado como parâmetro. Retorna zero se o vetor está vazio.

Exemplos:

```
v[0] ← 1
v[0] ← 2
imprima tamanho(v)           # imprimiria 2
```

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 que será impresso pelo algoritmo a seguir?

```
função estouComSorte(entradas: chave, URL[], texto[], listaNegra, rank[])  
início  
    melhorPagina ← 0  
    para i ← 1 até tamanho(texto) faça  
        se (find(texto[i], chave) <> -1) E (find(listaNegra, URL[i]) = -1)  
            E ((melhorPagina = 0) OU (rank[i] > rank[melhorPagina])) então  
                melhorPagina ← i  
        fim se  
    próximo i  
    resultado ← melhorPagina  
fim  
  
início  
    chave ← "casa"  
    URL[1] ← "http://a.se"  
    URL[2] ← "http://b.se"  
    texto[1] ← "casa baixa"  
    texto[2] ← "casa alta"  
    rank[1] ← 20  
    rank[2] ← 1  
    listaNegra ← "http://a.se;http://c.se"  
    idx ← estouComSorte(chave, URL, texto, listaNegra, rank)  
    imprima URL[idx]  
fim
```

- A) casa alta
- B) http://a.se
- C) http://b.se
- D) http://c.se
- E) Nenhuma das respostas anteriores

2ª questão (valor 1.0)

Determine o que será impresso pelo algoritmo a seguir. Assuma que o comando **imprima** não provoca uma mudança de linha após a execução.

```
procedimento encontraCelebridades(entradas: pessoa[], grafo[])  
início  
    para i ← 1 até tamanho(pessoa) faça  
        quantosMeConhecem ← 0  
        para j ← 1 até tamanho(grafo) faça  
            se find(grafo[j], pessoa[i]) != -1 então  
                quantosMeConhecem ← quantosMeConhecem + 1  
            fim se  
        próximo j  
        se quantosMeConhecem >= tamanho(grafo)/2 então  
            imprima pessoa[i]  
        fim se  
    próximo i  
fim
```

início

```
    pessoa[1] ← "Bruno"
    pessoa[2] ← "Angela"
    pessoa[3] ← "Camila"
    pessoa[4] ← "David"
    grafo[1] ← "Angela;Camila;David"
    grafo[2] ← "David"
    grafo[3] ← "Angela"
    grafo[4] ← "Angela"
    encontraCelebriedades(pessoa, grafo)
```

fim

- A) **Angela David**
- B) **Angela Bruno Camila David**
- C) **Angela**
- D) **Bruno**
- E) Nenhuma das respostas anteriores

3ª questão (valor 1.0)

Determine o que será impresso pelo algoritmo a seguir. Assuma que o comando **imprima** não provoca uma mudança de linha após a execução.

variáveis globais: cache[]

procedimento put(**entradas:** valor)

início

```
    se tamanho(cache) = 3 então
        cache[1] ← cache[2]
        cache[2] ← cache[3]
        cache[3] ← valor
    senão
        cache[tamanho(cache) + 1] ← valor
    fim se
```

fim

início

```
    put("Fantastica")
    put("Excelente")
    put("Extraordinaria")
    put("Brilhante")
    put("Incrível")
    para i ← 1 até tamanho(cache) faça
        imprima cache[i]
    próximo i
```

fim

- A) **Fantástica Excelente Incrível**
- B) **Fantástica Excelente Extraordinária**
- C) **Fantástica Excelente Extraordinária Brilhante Incrível**
- D) **Extraordinária Brilhante Incrível**
- E) Nenhuma das respostas anteriores

4ª questão (valor 1.0)

O que será impresso pelo algoritmo a seguir?

```
função ehIsograma(entradas: palavra)
início
    resultado ← verdadeiro
    para i ← 1 até 26 faça
        usadas[i] ← falso
    próximo i
    para i ← 1 até tamanho(palavra) faça
        ch ← charAt(palavra, i)
        se usadas[ordem(ch) - ordem('a') + 1] = verdadeiro então
            resultado ← falso
        fim se
        usadas[ordem(ch) - ordem('a') + 1] ← verdadeiro
    próximo i
fim

início
    imprima ehIsograma("flamengo"), ehIsograma("fluminense")
fim
```

A saída do algoritmo será:

- A) **falso falso**
- B) **falso verdadeiro**
- C) **verdadeiro falso**
- D) **verdadeiro verdadeiro**
- E) Nenhuma das respostas anteriores

5ª questão (valor 1.0)

O que será impresso pelo algoritmo a seguir?

```
função f1(entradas: p1, p2, p3)
início
    p1 ← p1 + p2 - 3
    p3 ← p3 + 1
    resultado ← p3 - p1
fim

função f2(entradas: p4, p5, p6)
início
    aux ← p4 / p5
    p6 ← aux * 2
    p5 ← p4 + 3 / p4 - 3
    resultado ← f1(7, p5, aux)
fim

início
    p3 ← 2
    p2 ← 3 + 2 * p3
    p1 ← f1(p3, p2, p3)
    imprima p1, f2(p3, p2, p1)
fim
```

- A) -2 2
- B) -3 -3
- C) -7 -2
- D) 9 -2
- E) Nenhuma das respostas anteriores

6ª questão (valor 1.0)

O que será impresso pelo algoritmo a seguir? Considere que o comando **imprima** não muda de linha após a impressão.

```
início
  para i ← 0 até 10 faça
    se (i mod 3 = 0) E (i mod 5 = 0) então
      imprima "a"
    senão
      se i mod 3 = 0 então
        imprima "b"
      senão
        se i mod 5 = 0 então
          imprima "c"
        fim se
      fim se
    fim se
  próximo i
fim
```

- A) a b c b b c
- B) b a c b c a
- C) c c a a a a
- D) c b a c c a
- E) Nenhuma das respostas anteriores

7ª questão (valor 1.0)

O que será impresso pelo algoritmo a seguir? Considere que o comando **imprima** não muda de linha após a impressão.

```
início
  imprima 5.0 + 1 / 2 * 17 mod 3
  imprima (4 <> 9 * 2) E (7 mod 2 = 1) OU (4 / 3 < 1)
fim
```

- A) 5.0 falso
- B) 7.5 verdadeiro
- C) 5.0 verdadeiro
- D) 7.5 falso
- E) Nenhuma das respostas anteriores

8ª questão (valor 1.0)

Assinale a opção a seguir que contém um nome de variável válido.

- A) 2coisas
- B) media aritmetica
- C) __idade
- D) d3b1
- E) Nenhuma das respostas anteriores

9ª questão (valor 1.0)

A principal característica do laço **enquanto** é:

- A) As instruções no laço são executadas um número infinito de vezes
- B) A condição do laço é testada antes do corpo do laço ser executado
- C) As instruções no laço são executadas pelo menos uma vez
- D) As instruções no laço são repetidas um número fixo de vezes
- E) Nenhuma das respostas anteriores

10ª questão (valor 1.0)

O que está errado com o algoritmo a seguir?

```
início
    soma ← 0
    enquanto soma <= 1000 faça
        soma ← soma - 30
    fim enquanto
    imprima soma
fim
```

- A) Deveria haver um ponto e vírgula ao final da instrução `soma ← soma - 30`
- B) A instrução dentro do laço deveria ser escrita como: `soma ← (soma - 30)`
- C) O teste da condição `soma <= 1000` deveria estar entre parênteses
- D) O laço enquanto nunca chegará ao final
- E) Nenhuma das respostas anteriores