

Curso de Tecnologia em Sistemas de Computação
Disciplina: Projeto e Desenvolvimento de Algoritmos
AP3 1º 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 checked="" type="checkbox"/>	D	E
2	A	B	<input checked="" type="checkbox"/>	D	E
3	A	B	C	<input checked="" type="checkbox"/>	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	<input checked="" type="checkbox"/>	C	D	E
8	A	<input checked="" type="checkbox"/>	C	D	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 `charAt()`, `substring()` e `tamanho()`, cuja documentação é mostrada a seguir:

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 ordem(entradas: car)

Retorna o valor ASCII do caracter `car`

Exemplo:

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

função str(entradas: num)

Retorna a string correspondente ao número passado como parâmetro

Exemplo:

```
imprima str(65) // imprimiria "65"
```

1ª questão (valor 1.0)

O que será impresso pelo algoritmo a seguir?

```
função tri(entradas: x, a, b, c)
início
    res ← 0.0
    se (x >= a) E (x <= b) então
        res ← (x-a)/(b-a)
    senão
        se (x > b) E (x <= c) então
            res ← (c-x)/(c-b)
        fim se
    fim se
    resultado ← res
fim

início
    imprima tri(2.75, 2.0, 3.0, 4.0)
fim
```

- A) 2.75
- B) 2.0
- C) 0.75
- D) 3.25
- E) Nenhuma das respostas anteriores

2ª questão (valor 1.0)

Nesta questão considere que o comando **imprima** não muda de linha ao final do comando. Considere também a existência de um comando **imprimaln** que faz com que o cursor mude para a próxima linha.

Observe o algoritmo a seguir.

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

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

procedimento trocal(entradas: tam, saídas vetor[])
início
    para i ← 1 até tam faça
        temp ← vetor[i]
        vetor[i] ← vetor[tam-i+1]
        vetor[tam-i+1] ← temp
    próximo i
fim
```

```

procedimento troca2(entradas: tam, saídas: vetor[])
início
    para i ← 1 até ≤ tam/2 faça
        temp ← vetor[i]
        vetor[i] ← vetor[tam-i+1]
        vetor[tam-i+1] ← temp
    próximo i
fim

início
    le(5, vetor)
    troca1(5, vetor)
    imprime(5, vetor)
    troca2(5, vetor)
    imprime(5, vetor)
fim

```

Se os valores fornecidos ao algoritmo forem 1, 2, 3, 4 e 5 nesta ordem, os valores impressos serão:

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

3ª questão (valor 1.0)

Observe o algoritmo a seguir:

```

função s(entradas: n, p)
início
    r ← 0
    para i ← 1 até p faça
        r ← r + n
    próximo i
    resultado ← r
fim

função e(entradas: n, p)
início
    r ← 1
    para i ← 1 até p faça
        r ← r * n
    próximo i
    resultado ← r
fim

início
    n ← 3
    p ← 5
    imprima s(n,p), ' ', e(n,p)
fim

```

Este algoritmo irá imprimir os seguintes valores:

- A) 3 5
- B) 5 3
- C) 35 53
- D) 15 243
- E) Nenhuma das respostas anteriores

4ª questão (valor 1.0)

Observe o algoritmo a seguir:

```
procedimento sp(entradas: tam, v[], saídas: vp[], vi[])  
início  
    ip ← 1  
    ii ← 1  
    para i ← 1 até tam faça  
        se (v[i] mod 2 = 0) então  
            vp[ip] ← v[i]  
            ip ← ip + 1  
        senão  
            vi[ii] ← v[i]  
            ii ← ii + 1  
        fim se  
    próximo i  
    vi[ii] ← -1  
    vp[ip] ← -1  
fim  
  
procedimento imprime(entradas: v[])  
início  
    i ← 1  
    enquanto (v[i] >= 0) faça  
        imprima v[i], ' '  
        i ← i + 1  
    fim enquanto  
fim  
  
início  
    v[1] ← 11          v[2] ← 15          v[3] ← 4  
    v[4] ← 13          v[5] ← 22  
    sp(5, v, vp, vi)  
    imprime(vp)  
    imprime(vi)  
fim
```

Este algoritmo irá imprimir os seguintes valores:

- A) 4 22 11 15 13
- B) 4 11 22 15 13
- C) 11 15 13 4 22
- D) 11 4 15 22 13
- E) Nenhuma das respostas anteriores

5ª questão (valor 1.0)

Observe o algoritmo a seguir:

```
início
    t ← 100.00
    p[1] ← 23.25    p[2] ← 42.5    p[3] ← 20.5
    p[4] ← 18.75    p[5] ← 15.50
    c ← 0.0
    i ← 0
    enquanto (c < t) E (i <= 5) faça
        i ← i + 1
        c ← c + p[i]
    fim enquanto
    se (c > t) então
        c ← c - p[i]
        i ← i - 1
    fim se
    imprima i, ' ', c
fim
```

Este algoritmo irá imprimir os seguintes valores:

- A) 5 120.5
- B) 3 86.25
- C) 1 23.25
- D) 4 105.00
- E) Nenhuma das respostas anteriores

6ª questão (valor 1.0)

Três propriedades da tabela ASCII merecem atenção especial:

- Os códigos de caracteres para os dígitos são consecutivos.
- As letras do alfabeto são divididas em duas faixas, uma para as letras maiúsculas e uma para as letras minúsculas. Dentro de cada faixa, os valores ASCII são consecutivos.
- O número zero não é a mesma coisa que o caractere zero. O mesmo vale para todos os dígitos.

Assim, o valor impresso pelo algoritmo a seguir, é:

```
início
    imprima concat(str((ordem('c') - ordem('a')) + 3), "a")
fim
```

- A) 6765a
- B) 99973a
- C) 23a
- D) 5a
- E) Nenhuma das respostas anteriores

7ª questão (valor 1.0)

No algoritmo a seguir, que altura resulta em um grau de certeza de aproximadamente 0.3 de que uma pessoa é alta?

```

início
  leia altura
  se altura < 5 então
    grauCerteza ← 0
  senão
    se altura >= 1.52 e altura <= 2.13 então
      grauCerteza ← (altura - 1.52)/0.61
    senão
      grauCerteza ← 1
    fim se
  fim se
  imprima grauCerteza
fim

```

- A) 1.65
- B) 1.70
- C) 1.75
- D) 1.80
- E) 1.85

8ª questão (valor 1.0)

O que será impresso pelo algoritmo a seguir:

```

início
  numItens ← 12
  dados[1] ← 4
  dados[2] ← 4
  dados[3] ← 5
  dados[4] ← 5
  dados[5] ← 6
  dados[6] ← 6
  dados[7] ← 6
  dados[8] ← 7
  dados[9] ← 7
  dados[10] ← 7
  dados[11] ← 8
  dados[12] ← 8
  conta ← 0
  para i ← 1 até (numItens - 1) faça
    se dados[i] = dados[i + 1] então
      conta ← conta + 1
    fim se
  próximo i
  imprima conta
fim

```

- A) 6
- B) 7
- C) 8
- D) 9
- E) Nenhuma das respostas anteriores

9ª questão (valor 1.0)

```
início
  leia x
  y ← 0
  enquanto y < x faça
    y ← y + 7
  fim enquanto
fim
```

Assinale a alternativa verdadeira, depois que o laço **enquanto** no algoritmo acima tiver sido executado. Assuma que $x \geq 0$.

- A) y é necessariamente maior do que x
- B) y pode ser igual a $(x + 7)$
- C) y é necessariamente maior do que zero
- D) y pode ser igual a x
- E) Nenhuma das respostas anteriores

10ª questão (valor 1.0)

O vetor **dados** contém **numItens** elementos. O algoritmo a seguir deveria converter o vetor **dados** de:

7	3	8	1	0	5
---	---	---	---	---	---

em:

3	8	1	0	5	7
---	---	---	---	---	---

```
início
  // comando 1
  para i ← 1 até (numItens - 1) faça
    dados[i] ← dados[i + 1]
  próximo i
  // comando 2
fim
```

De modo a que o algoritmo execute corretamente, “comando 1” e “comando 2” deveriam ser:

- A)

```
temp ← dados[1]
dados[1] ← temp
```
- B)

```
temp ← dados[1]
dados[numItens] ← temp
```
- C)

```
temp ← dados[numItens]
dados[1] ← temp
```
- D)

```
temp ← dados[numItens]
dados[numItens] ← temp
```
- E) Nenhuma das respostas anteriores