



Fundação CECIERJ - Vice Presidência de Educação Superior a Distância

**Curso de Tecnologia em Sistemas de Computação**  
**Disciplina: Projeto e Desenvolvimento de Algoritmos**  
**AP3 2º semestre de 2007.**

**Nome –**

**Assinatura –**

---

Observações:

1. Prova sem consulta e sem uso de máquina de calcular.
  2. Use caneta para preencher o seu nome e assinar nas folhas de questões e nas folhas de respostas.
  3. Você pode usar lápis para responder as questões.
  4. Ao final da prova devolva as folhas de questões e as de respostas.
  5. Todas as respostas devem ser transcritas no local apropriado, ao final do caderno de questões.
- 

**1ª questão (valor 1.0)**

O que será impresso pelo algoritmo a seguir?

Observações:

- Considere que o comando *imprima* não muda automaticamente de linha após a impressão

**inicio**

```
A[1] ← 10
A[2] ← 3
A[3] ← 8
A[4] ← 1
A[5] ← 7
para i ← 2 até 5 faça
    value ← A[i]
    j ← i-1
    enquanto (j >= 1) e (A[j] > value) faça
        A[j + 1] ← A[j]
        j ← j-1
    fim enquanto
    A[j+1] ← value
próximo i
para i ← 1 até 5 faça
    imprima A[i], ' '
próximo i
```

**fim**

- A) 1 3 7 8 10
- B) 1 10 7 8 3
- C) 10 3 8 1 7
- D) 10 8 7 3 1
- E) 1 3 7 10 8

**2ª questão (valor 1.0)**

O que será impresso pelo algoritmo a seguir?

Observações:

- considere que o comando *imprima* não muda automaticamente de linha após a impressão

**procedimento** m(entradas: a[], b[]; saídas: result[])  
**início**

    i ← 1

    j ← 1

    k ← 1

**enquanto** (a[i] >= 0) e (b[j] >= 0) **faça**

**se** a[i] < b[j] **então**

            result[k] ← a[i]

            i ← i + 1

            k ← k + 1

**senão**

            result[k] ← b[j]

            j ← j + 1

            k ← k + 1

**fim se**

**fim enquanto**

**enquanto** (a[i] >= 0) **faça**

        result[k] ← a[i]

        i ← i + 1

        k ← k + 1

**fim enquanto**

**enquanto** (b[j] >= 0) **faça**

        result[k] ← b[j]

        j ← j + 1

        k ← k + 1

**fim enquanto**

    result[k] ← -1

**fim**

**procedimento** p(entradas: v[])

**início**

    i ← 1

**enquanto** (v[i] >= 0) **faça**

**imprima** v[i], ' '

        i ← i + 1

**fim enquanto**

**fim**

**início**

```
v1[1] ← 1
v1[2] ← 3
v1[3] ← 5
v1[4] ← 9
v1[5] ← -1
v2[1] ← 4
v2[2] ← 6
v2[3] ← 8
v2[4] ← -1

m(v1, v2, v3)
p(v3)
```

**fim**

- A) 1 3 5 9 -1
- B) 4 6 8 -1
- C) 1 3 5 9 -1 4 6 8 -1
- D) 1 3 4 5 6 8 9
- E) 1 3 4 5 6 8 9 -1

### 3ª questão (valor 1.0)

O que será impresso pelo algoritmo a seguir?

Considere que o usuário digitou a seguinte cadeia de caracteres:

**122344111**

Observações:

- Considere que o comando *imprima* não muda automaticamente de linha após a impressão
- Considere a existência de uma função chamada lerCadeia(vetor) que lê uma cadeia de caracteres para um vetor de caracteres qualquer (no exemplo de chamado vetor)
- Considere a existência de uma função chamada compr(vetor) que retorna o comprimento da cadeia de caracteres passada como parâmetro.

**início**

```
lerCadeia (vetor)
i ← 1
enquanto (i <= compr(vetor)) faça
    atual ← vetor[i]
    conta ← 1
    i ← i + 1
    enquanto (vetor[i] = atual) faça
        conta ← conta + 1
        i ← i + 1
fim enquanto
```

```

        imprima conta, atual
    fim enquanto
fim

```

- A) 122344111
- B) 12345678
- C) 1122132431
- D) 87654321
- E) 1223334444

#### 4ª questão (valor 1.0)

O que será impresso pelo algoritmo a seguir?

Observações:

- Considere que o comando **imprima** não muda automaticamente de linha após a impressão.
- Considere a existência de uma função chamada **imprimal** que ao terminar de imprimir o que foi pedido passa para a próxima linha.

```

início
    para i ← 0 até 4 faça
        para j ← 1 até 5 faça
            imprima i * j
        próximo j
        imprimal
    próximo i
fim

```

A)

0	0	0	0	0
1	2	3	4	5
2	4	6	8	10
3	6	9	1	15
			2	
4	8	1	1	20
		2	6	

B)

1	2	3	4	5
2	4	6	8	10
3	6	9	1	15
			2	
4	8	1	1	20
		2	6	

C)

0	0	0	0	0
1	2	3	4	5
2	4	6	8	10
3	6	9	1	15
			2	

D)

0	0	0	0
1	2	3	4
2	4	6	8
3	6	9	12
4	8	1	16
		2	

E)

1	2	3	4	5
2	4	6	8	10
3	6	9	1	15
			2	
4	8	1	1	20
		2	6	
0	0	0	0	0

- A) 122344111
- B) 12345678
- C) 1122132431
- D) 87654321
- E) 1223334444

**5ª questão (valor 1.0)**

O que será impresso pelo algoritmo a seguir?

Observações:

- Considere que o comando *imprima* não muda automaticamente de linha após a impressão

**variável global: i**

**função f1 (saídas: i)**

**início**

**imprima** i, ' '

$i \leftarrow i + 1$

**resultado** i / 2

**fim**

**função f2 (entradas: i)**

**início**

**imprima** i, ' '

$i \leftarrow i + 1$

**resultado** i / 2

**fim**

**procedimento f3 (entradas: i; saídas: r)**

**início**

**imprima** i, ' '

$i \leftarrow i + 1$

$r \leftarrow i$

**fim**

**início**

$i \leftarrow 15$

$i \leftarrow f1(i)$

**imprima** i, ' '

$i \leftarrow f2(i)$

**imprima** i, ' '

    f3(i, i)

**imprima** i, ' '

**fim**

- A) 16 8 8 4 4 5
- B) 15 8 9 4 4 5
- C) 15 8 8 4 5 5
- D) 15 8 8 4 4 5
- E) 16 8 8 4 5 5

### 6ª questão (valor 1.0)

O que será impresso pelo algoritmo a seguir?

Observações:

- Suponha a existência de um tipo **string** que pode ser tratado como um array de caracteres

Exemplo: O trecho de código,

```
s ← 'Alo'
s[1] ← 'F'
imprima s
```

imprimiria  
Flo

- Considere a existência da função **tamanho(str)** que devolve o tamanho da **string** str

Exemplo: O trecho de código,

```
str ← 'Alo'
imprima tamanho(str)
```

imprimiria  
3

Algoritmo:

**função** replace(**saídas**: sat)

**início**

**resultado** := 0

    dim ← tamanho(sat)

**para** i ← 1 **até** dim **faça**

**se** sat[i]=' ' **então**

            j ← tamanho(sat)

**enquanto** j >= i+1 **faça**

                sat[j] ← sat[j-2]

                j ← j - 1

**fim enquanto**

            sat[i] ← '\*'

            sat[i] ← 'h'

            sat[i+1] ← '+'

            sat[i+1] ← 'a'

            sat[i+2] ← '#'

            sat[i+2] ← 't'

**resultado** ← **resultado** + 1

**fim se**

**próximo** i

**fim**

**início**

    cat ← 'The cat sat'

    n ← replace(cat)

```
    imprima cat
fim
```

- A) The#cat#sat
- B) The\*cat\*sat
- C) The+cat+sat
- D) Thehatcathatsat
- E) Thehcatasatt

**7ª questão (valor 1.0)**

O que será impresso pelo algoritmo a seguir?

Considere que a entrada digitada pelo usuário foi:

```
200
38
220
48
240
68
0
```

Algoritmo

```
função calculaPagamento(entradas: taxa, horas)
início
```

```
    resultado ← 0
    se horas > 60 então
        resultado ← resultado + (horas-60)*2*taxa
        horas ← 60
    fim se
    se horas > 40 então
        resultado ← resultado + (horas-40)*1.5*taxa
        horas ← 40
    fim se
    resultado ← (resultado + horas*taxa)/100
```

```
fim
```

```
início
```

```
    total ← 0
    acabou ← falso
    enquanto não acabou faça
        imprima 'taxa (centavos/hora): '
        leia taxa
        se taxa=0 então
            acabou ← verdadeiro
        senão
            imprima 'horas: '
            leia horas
            total ← total + calculaPagamento(taxa, horas)
        fim se
    fim enquanto
    imprima total
fim
```

- A) 140.80
- B) 396.80
- C) 560.80

- D) 597.60  
E) 651.90

**8ª questão (valor 1.0)**

O que será impresso pelo programa a seguir?

Considere a existência do tipo **string** e da função **tamanho()** definidas na 6ª questão. Considere ainda que a função **imprima** não muda de linha a cada impressão. Algoritmo:

```
início
  para i ← 1 até 10 faça
    contaPalavras[i] ← 0
  próximo i

  str ← 'Minha terra tem palmeiras onde canta o sabiá'
  contaLetras ← 0

  para i ← 1 até tamanho(str) faça
    se str[i]=' ' então
      contaPalavras[contaLetras] ← contaPalavras[contaLetras] + 1
      contaLetras ← 0
    senão
      contaLetras ← contaLetras + 1
    fim se
  próximo i

  contaPalavras[contaLetras] ← contaPalavras[contaLetras] + 1
  para i ← 1 até 10 faça
    imprima contaPalavras[i]
  próximo i
fim
```

- A) 0000000000  
B) 1011100010  
C) 1011400010  
D) 1114100000  
E) 1234567890

**9ª questão (valor 1.0)**

O que será impresso pelo algoritmo a seguir?

Considere que a entrada digitada pelo usuário foi o número 9 (nove)

Algoritmo

```
início
  conta ← 0
  leia n
  enquanto n <> 1 faça
    conta ← conta + n
    se n mod 2 = 0 então
      n ← n / 2
    senão
      n ← 3*n+1
```



```

        fim se
    fim enquanto
    imprima conta+1
fim

```

- A) 15
- B) 46
- C) 119
- D) 259
- E) 339

### 10ª questão (valor 1.0)

O que será impresso pelo algoritmo a seguir?

Considere que, quando solicitado, o usuário digita como entrada os números inteiros 8 (oito) e 77 (setenta e sete), nesta ordem.

Algoritmo

```

função pow (entradas: base, exp)
início
    resultado ← 1
    para i ← 1 até exp faça
        resultado ← resultado * base
    próximo i
fim

início
    leia base
    leia num
    i ← 0
    saída ← 0
    enquanto num>0 faça
        aux ← num mod 10
        saída ← saída + aux*pow(base, i)
        num ← num/10
        i ← i + 1
    fim enquanto
    imprima saída
fim

```

- A) 63
- B) 77
- C) 415
- D) 616
- E) 1001101

Questão					
1	A	B	C	D	E
2	A	B	C	D	E
3	A	B	C	D	E
4	A	B	C	D	E
5	A	B	C	D	E
6	A	B	C	D	E
7	A	B	C	D	E
8	A	B	C	D	E
9	A	B	C	D	E
10	A	B	C	D	E

## Gabarito

1. C
2. E
3. A
4. A
5. C
6. D
7. B
8. E
9. C
10. C