



Fundação CECIERJ - Vice Presidência de Educação Superior a Distância

**Curso de Tecnologia em Sistemas de Computação**  
**Disciplina: Projeto e Desenvolvimento de Algoritmos**  
**AD2 1º semestre de 2019.**

Nome –

Assinatura –

---

## **1ª questão (valor 5.0)**

Escreva um algoritmo que descubra as respostas de um aluno a uma prova de múltipla escolha. Nesta prova, cada questão terá cinco alternativas (A, B, C, D e E). O aluno escolhe a resposta preenchendo retângulos. Ao final da prova, as folhas de respostas serão escaneadas e processadas digitalmente.

A prova foi escaneada usando tons de cinza entre 0 (preto total) e 255 (branco total). O scanner após detectar os cinco retângulos reservados para as respostas de uma questão retorna cinco valores inteiros correspondentes a cada uma das respostas. Se o quadrado foi preenchido corretamente o valor da média é zero (preto total). Se o quadrado foi deixado em branco o valor da média é 255 (branco total). Assim, idealmente, se os valores de cada quadrado de uma questão são (255, 0, 255, 255, 255), sabemos que o aluno marcou a alternativa B para essa questão.

Obviamente, nem todas as questões das folhas de resposta são marcadas de maneira correta. Pode acontecer de um aluno se enganar e marcar mais de uma alternativa na mesma questão, ou não marcar nenhuma alternativa. Nesses casos, a resposta deve ser desconsiderada e impresso um asterisco.

Escreva um algoritmo que leia para cada questão os valores dos cinco retângulos de uma questão e imprima a letra marcada, ou se a resposta à questão deve ser desconsiderada e impresso um asterisco.

**Importante: o primeiro dado a ser lido de uma prova é a quantidade de questões da prova.**

A seguir mostramos dois exemplos de execuções diferentes de um possível algoritmo. Em negrito estão os textos impressos pelo computador.

Exemplo 1:

**Quantas questoes? 2**

**Questao 1**

? 255

? 255

? 0

? 255

? 255

**Resposta C**

**Questao 2**

? 0

? 255

? 255

? 255

? 255

**Resposta A**

Exemplo 2:

**Quantas questoes? 3**

**Questao 1**

? 255

? 0

? 255

? 255

? 255

**Resposta B**

**Questao 2**

? 0

? 255

? 255

? 0

? 255

**Resposta \***

**Questao 3**

? 255

? 255

? 255

? 255

? 0

**Resposta E**

```

início
    letras[1] ← 'A'
    letras[2] ← 'B'
    letras[3] ← 'C'
    letras[4] ← 'D'
    letras[5] ← 'E'
    escreva "Quantas questoes? "
    leia total_questoes
    para j ← 1 até total_questoes faça
        imprima "Questao ", j
        selecionadas ← 0
        para i ← 1 até 5 faça
            imprima "? "
            leia cor_resposta
            se cor_resposta = 0 então
                resposta ← i
                selecionadas ← selecionadas + 1
            fim se
        próximo i
        se selecionadas > 1 então
            imprima "Resposta *"
        senão
            se selecionadas = 0 então
                imprima "Resposta *"
            senão
                imprima "Resposta ", letras[resposta]
            fim se
        fim se
    próximo j
fim

```

## 2ª questão (valor 5.0)

Escreva um procedimento PETEQS chamado **iniciais(entradas: lista[], k)** que recebe dois parâmetros como entrada: um vetor de strings e um número inteiro **k**. Seu procedimento deve examinar as strings no vetor recebido e imprimir as letras que são as iniciais de pelo menos **k** strings no vetor. Em outras palavras, se **k** ou mais strings no vetor começarem com a mesma letra, essa letra deverá fazer parte do conjunto de letras impressas. Você pode considerar que todas as palavras no vetor de entrada são compostas apenas por letras minúsculas.

Exemplo: Considere um vetor de entrada chamado **lista** contendo as seguintes strings:

```
lista = ["quando", "vou", "bar", "em", "bar", "viro", "a",  
        "mesa", "berro", "bebo", "e", "brigo", "que",  
        "viagem"]
```

Cinco palavras na lista começam com "**b**", três com "**v**", duas com "**e**", duas com "**q**", uma com "**a**" e uma com "**m**". Portanto, a chamada **iniciais(lista, 2)**, deveria imprimir:

```
b  
e  
q  
v
```

A chamada **iniciais(lista, 3)**, deveria imprimir:

```
b  
v
```

As chamadas **iniciais(lista, 4)**, ou **iniciais(lista, 5)**, deveriam imprimir:

```
b
```

A chamada **iniciais(lista, 6)** não deveria imprimir nenhuma letra.

Dica: Penso que ajudaria criar em sua função a variável local **alfabeto**. Algo como:

```
procedimento iniciais(entradas: lista[], k)  
início  
    alfabeto ← ['a', 'b', 'c', 'd', 'e',  
               'f', 'g', 'h', 'i', 'j',  
               'k', 'l', 'm', 'n', 'o',  
               'p', 'q', 'r', 's', 't',  
               'u', 'v', 'w', 'x', 'y', 'z']
```

*# o restante do seu código vai aqui...*

Fim

Em sua solução, considere a existência em PETEQS das funções **tamanho()** e **charAt()** cuja documentação é mostrada a seguir:

**função tamanho(entradas: v[])**

Retorna o número de elementos no vetor **v** recebido como parâmetro.

Exemplos:

```
imprima tamanho(["quando", "vou", "bar", "em", "bar"])  
# imprimiria 5
```

**função charAt(entradas: str, pos)**

Retorna o caractere na posição **pos** da cadeia de caracteres **str** passada como parâmetro.

Exemplo:

```
imprima charAt("CEDERJ", 3)
```

# imprimiria D

```
procedimento iniciais(entradas: lista[], k)
início
    alfabeto ← ['a', 'b', 'c', 'd', 'e',
                'f', 'g', 'h', 'i', 'j',
                'k', 'l', 'm', 'n', 'o',
                'p', 'q', 'r', 's', 't',
                'u', 'v', 'w', 'x', 'y', 'z']
    para i ← 1 até tamanho(alfabeto) faça
        letra ← alfabeto[i]
        conta ← 0
        para j ← 1 até tamanho(lista) faça
            se charAt(lista[j], 1) = letra então
                conta ← conta + 1
            fim se
        próximo j
        se conta >= k então
            imprima letra
        fim se
    próximo i
fim
```