

## Gabarito proposto

### Introdução à POO (INE5603)

Sistemas de Informação – Universidade Federal de Santa Catarina

Estudante: \_\_\_\_\_

1. (2,5) Escreva um método “mostraPalavrasComCaracter” que receba um vetor de *String* e um caracter, e mostre, na tela, apenas as palavras do vetor que tenham este caracter. **Dicas:** considerando um string “s”, “s.length()” é sua quantidade total de caracteres, e “s.charAt(i)” acessa seu caracter no índice i. Segue uma sugestão de organização do código:

```
public class Prova {
    public static void mostraPalavrasComCaracter(String[] vetor, char caracter) {
        // COLOCAR SEU CÓDIGO AQUI
    }
    public static void main(String[] args) {
        String[] palavras = {"Queijo", "Tomate", "Azeitonas Pretas", "Peru", "Massa fina"};
        mostraPalavrasComCaracter(palavras, 'a');
    }
}
```

Para o exemplo de “main” acima, a execução deve listar as palavras que contêm o caracter “a” (minúsculo):  
Tomate, Azeitonas Pretas, Massa fina

```
public static void mostraPalavrasComCaracter(String[] vetor, char caracter) {
    for (String s : vetor) {
        boolean E = false;
        for (int i=0; i < s.length(); i++) {
            if (s.charAt(i) == caracter) {
                E = true;
                break;
            }
        }
        if (E)
            System.out.println(s);
    }
}
```

2. (2,5) Uma lista de elementos é representada por um vetor  $v$  de inteiros, na qual o conteúdo de uma posição é o valor  $e$ , ao mesmo tempo, o índice do próximo elemento. A posição 0 sempre é considerada início da lista. E o último elemento da lista é a primeira ocorrência do valor de próximo como  $-1$ . Por exemplo, o vetor a seguir contém a lista “0  $\rightarrow$  4  $\rightarrow$  2  $\rightarrow$  5  $\rightarrow$  3”, pois no índice 0 há o valor 4; no índice 4 há o valor 2; no índice 2 há o valor 5; no índice 5 há o valor 3; e no índice 3 há o valor -1:

4	-1	5	-1	2	3
0	1	2	3	4	5

Escreva um método, em Java, que receba um vetor de inteiros neste formato (de qualquer tamanho) e retorne o número de elementos da lista representada.

```

public class Q2 {
    public static int numeroElementos(int[] vetor) {
        int numero = 0;
        int indice = 0;
        while (vetor[indice] != -1) {
            indice = vetor[indice];
            numero++;
        }
        return numero;
    }

    public static void main(String[] args) {
        int[] vetor = {4, -1, 5, -1, 2, 3};
        int numero = numeroElementos(vetor);
        System.out.println("numero: " + numero);
    }
}

```

3. (2,5) Uma imagem binária, de tamanho  $m \times n$ , pode ser representada por uma matriz “img”, na qual cada elemento assume ou valor 0 (preto) ou valor 1 (branco). Veja um exemplo de declaração (com alocação e atribuição) em Java:

```

int img[][] = { {1, 1, 1, 1, 1, 1, 1, 1},
                {0, 1, 0, 1, 1, 1, 0, 0},
                {0, 0, 0, 0, 0, 0, 0, 0},
                {1, 1, 1, 1, 1, 1, 1, 1},
                {1, 1, 0, 0, 1, 1, 1, 0} };

```

Escreva um método que receba uma imagem de inteiros (como a do exemplo) e devolva um vetor de inteiros com quatro informações sobre uma imagem binária:

- A quantidade de pixels pretos;
- A quantidade de pixels brancos;
- A quantidade de linhas totalmente pretas;
- A quantidade de linhas totalmente brancas.

```

public class Q3 {

    public static int quantPixelsPretos(int[][] img) {
        int soma = 0;
        for (int i=0; i < img.length; i++)
            for (int j=0; j < img[i].length; j++)
                if (img[i][j] == 0)
                    soma++;
        return soma;
    }

    public static int quantPixelsBrancos(int[][] img) {
        int soma = 0;
        for (int i=0; i < img.length; i++)
            for (int j=0; j < img[i].length; j++)
                if (img[i][j] == 1)
                    soma++;
    }
}

```

```

        return soma;
    }

    public static int quantLinhasPretas(int[][] img) {
        int soma = 0;
        for (int i=0; i < img.length; i++) {
            boolean T = true;
            for (int j=0; j < img[i].length; j++)
                if (img[i][j] != 0) {
                    T = false;
                    break;
                }
            if (T)
                soma++;
        }
        return soma;
    }

    public static int quantLinhasBrancas(int[][] img) {
        int soma = 0;
        for (int i=0; i < img.length; i++) {
            boolean T = true;
            for (int j=0; j < img[i].length; j++)
                if (img[i][j] != 1) {
                    T = false;
                    break;
                }
            if (T)
                soma++;
        }
        return soma;
    }

    public static void main(String[] args) {
        int img[][] = { {1, 1, 1, 1, 1, 1, 1, 1},
                        {0, 1, 0, 1, 1, 1, 0, 0},
                        {0, 0, 0, 0, 0, 0, 0, 0},
                        {1, 1, 1, 1, 1, 1, 1, 1},
                        {1, 1, 0, 0, 1, 1, 1, 0} };
        System.out.println("quantidade de pixels pretos : " + quantPixelsPretos(img));
        System.out.println("quantidade de pixels brancos: " + quantPixelsBrancos(img));
        System.out.println("quantidade de linhas pretas : " + quantLinhasPretas(img));
        System.out.println("quantidade de linhas brancas: " + quantLinhasBrancas(img));
    }
}

```

4. (2,5) Escreva um vetor ordenado (ordem crescente) contendo ao menos 10 inteiros. Explique a busca

---

binária pela posição (índice) de um dado valor, demonstrando seu funcionamento passo-a-passo no vetor criado. Explique o ganho obtido em relação a uma busca linear.