

Nome:

RA:

Instruções para a prova (leia antes de começar):

- 1) A prova tem a duração de 1h50min.
- 2) A prova é individual.

Questão 1 (6 pontos)

Escreva uma função que permita juntar (concatenar) 3 vetores de números inteiros

Definição da função: `static int[] concatenar (int L1[], int L2[], int L3[])`

Exemplo:

Para os vetores `L1=[11, 22]`, `L2=[55, 44, 11]`, `L3=[88, 99, 22, 44]` a resposta deve ser `[11, 22, 55, 44, 11, 88, 99, 22, 44]`.

Alternativa 1:

```
static int[] concatenar1 (int L1[], int L2[], int L3[]) {
    int R[] = new int[L1.length + L2.length + L3.length];
    int i;

    for (i=0; i<L1.length; i=i+1)
        R[i] = L1[i];
    for (i=0; i<L2.length; i=i+1)
        R[i+L1.length] = L2[i];
    for (i=0; i<L3.length; i=i+1)
        R[i+L1.length+L2.length] = L3[i];

    return R;
}
```

Alternativa 2:

```
static int[] concatenar2 (int L1[], int L2[], int L3[]) {
    int n = L1.length + L2.length + L3.length;
    int R[] = new int[n];
    int i;

    for (i=0; i<n; i=i+1) {
        if (i<L1.length) {
            R[i] = L1[i];
        }
        else {
            if (i<L1.length+L2.length) {
                R[i] = L2[i-L1.length];
            }
            else {
                R[i] = L3[i-L1.length-L2.length];
            }
        }
    }
    return R;
}
```

Questão 2 (4 pontos)

Escreva uma função que permita imprimir os 100 primeiros números inteiros, **pares** maiores que zero e múltiplos de 3 e de 5.

Definição da função: static void imprimirNumeros()

```
static void imprimirNumeros() {
    int i, cont=0;

    for(i=1; cont<100; i=i+1) {
        if (i%2==0 && i%3==0 && i%5==0) {
            cont = cont+1;
            System.out.printf("%d - %d\n", cont, i);
        }
    }
}
```

Questão 3 (6 pontos)

Qual é o resultado da execução do seguinte método?

```
static void triangulo () {
    int i, j;
    int M[][] = new int[6][6];

    for (i=5; i>=0; i=i-1) {
        for (j=i; j<6; j=j+1) {
            if (i==j) {
                M[i][j] = 1;
            }
            else {
                M[i][j] = M[i+1][j] + M[i][j-1];
            }
        }
    }

    for (i=0; i<6; i=i+1) {
        for (j=0; j<6; j=j+1) {
            System.out.print(M[i][j]+" ");
        }
        System.out.print("\n");
    }
}
```

Solução:

```
1  2  4  8 16 32
0  1  2  4  8 16
0  0  1  2  4  8
0  0  0  1  2  4
0  0  0  0  1  2
0  0  0  0  0  1
```

Questão 4 (6 pontos)

Escreva uma função que permita verificar se existem pelo menos 5 números iguais em um vetor de números inteiros.

Definição da função: `static boolean cinco(int v[])`

Exemplos:

Para o vetor [11, 22, 11, 33] a resposta deve ser false

Para o vetor [11, 22, 11, 22, 33, 44, 55, 22, 66, 22, 22] a resposta deve ser true

Para o vetor [55, 55, 55, 55, 55] a resposta deve ser true

```
static boolean cinco( int v[] ) {
    int i, j, cont;

    for (i=0; i<v.length-1; i=i+1) {
        cont = 1;

        for (j=i+1; j<v.length; j=j+1) {
            if (v[i]==v[j]) {
                cont = cont+1;
            }
        }

        if (cont>=5) {
            return true;
        }
    }

    return false;
}
```

Questão 5 (6 pontos)

Crie uma função para determinar o menor valor presente na diagonal secundária de uma matriz (quadrada ou retangular) dada como entrada.

Definição da função: `static int menorValor(int M[][])`

Limitação: Sua implementação deve ser eficiente e somente os elementos que estão na diagonal deverão ser consultados. Não serão aceitas soluções que considerem laços aninhados (apenas 1 laço é suficiente).

```
static int menorValor( int M[][] ) {
    int i, menor, p=M.length;

    if (p > M[0].length) {
        p = M[0].length;
    }
    menor = M[0][p-1];

    for (i=0; i<p; i=i+1) {
        if (menor > M[i][p-i-1]) {
            menor = M[i][p-i-1];
        }
    }

    return menor;
}
```

Questão 6 (6 pontos)

Escreva uma função que permita verificar se existem pelo menos duas linhas idênticas.

Definição da função: `static boolean linhasIdenticas(int M[][])`

```
static boolean linhasIdenticas ( int M[][] ) {
    int i, j, k, cont;
    int n = M.length;
    int m = M[0].length;

    for (i=0; i<n; i=i+1) {
        for (j=i+1; j<n; j=j+1) {
            cont=0;
            for (k=0; k<m; k=k+1) {
                if (M[i][k]==M[j][k]) {
                    cont = cont+1;
                }
            }
            if (cont==m) {
                return true;
            }
        }
    }

    return false;
}
```

Questão 7 (6 pontos)

Considere o método `particiona`. Avalie o método com os seguintes elementos

`p = 0`

`r = 9`

`A = [99, 33, 55, 77, 11, 22, 88, 66, 33, 44]`

```
static int particiona ( int A[], int p, int r ) {
    int i, j, x, aux;
    x = A[r];
    i = p-1;

    for (j=p; j<=r-1; j=j+1) {
        if (A[j]>=x) {
            i = i+1;
            aux = A[i];
            A[i] = A[j];
            A[j] = aux;
        }
    }

    aux = A[i+1];
    A[i+1] = A[r];
    A[r] = aux;

    return i+1;
}
```

(a) Qual a resposta do método (i.e., indique o número inteiro que o método devolve): [2 pontos]

5

(b) Apresente o vetor A após a execução do método. [4 pontos]

`A = [99, 55, 77, 88, 66, 44, 33, 11, 33, 22]`