

Nome:	RA:
-------	-----

**Instruções para a prova (leia antes de começar):**

- 1) A prova tem a duração de 1h50min.
- 2) A compreensão e interpretação do enunciado é parte integrante da avaliação. A prova é individual e sem consultas.

**Questão 1:** Selecione a opção correta. Resposta correta +2, incorreta -1 (10 pontos)

<pre>static boolean F1 (int p, int q) {     if (true)         return true;     if (p==q)         return true;     else         return false; }</pre>	<p>Para dois números inteiros p e q, a função devolve:</p> <p>(a) true se p é igual a q.  (b) false se p é igual a q.  <b>(c) true independente dos valores de p e q.</b>  (d) false independente dos valores de p e q .  (e) Nenhuma das anteriores.</p>
<pre>static int numeros1 (int x, int y, int z) {     if (x&lt;y) {         if (y&lt;z)             return z;         else             return y;     }     else {         if (x&lt;z)             return z;         else             return x;     } }</pre>	<p>Para três números inteiros diferentes x, y, e z, a função devolve:</p> <p>(a) O menor valor entre x, y e z.  <b>(b) O maior valor entre x, y e z.</b>  (c) z e y.  (d) z e x .  (e) Nenhuma das anteriores.</p>
<pre>static void G1 (int n) {     int i;     for (i=0; i&lt;n*2; i=i+2)         System.out.println("SPAM"); }</pre>	<p>Para um número inteiro positivo n, a função:</p> <p><b>(a) Imprime n vezes a palavra “SPAM”.</b>  (b) Imprime n-1 vezes a palavra “SPAM”.  (c) Imprime 2n vezes a palavra “SPAM”.  (d) Imprime 2n-1 vezes a palavra “SPAM”.  (e) Nenhuma das anteriores.</p>
<pre>static int H( int v[ ] ) {     int i;     int n = v.length;     int soma = 0;      for (i=0; i&lt;n; i=i+1)         soma = 2*(soma + v[i]);      return soma; }</pre>	<p>Para v={3,4,2,1}, a função devolve</p> <p>(a) 80.  (b) 86.  <b>(c) 90.</b>  (d) 96.  (e) Nenhuma das anteriores.</p>
<pre>static String J( String p ) {     int i;     String r = "";      for (i=0; i&lt;p.length(); i=i+2) {         r = r + p.charAt(i);         r = r + p.charAt(i) + r;     }      return r; }</pre>	<p>Para p="abc", a função devolve</p> <p><b>(a) “aaacccaaac”.</b>  (b) “aaabbaaaab”.  (c) “bbbaabbba”.  (d) “bbbccbbbc”.  (e) Nenhuma das anteriores.</p>

**Questão 2:** (8 pontos)

Indique o valor que devolverá o seguinte método. Considere como parâmetro de entrada, para o método, o seu número de matrícula (isto é, seu RA composto de 8 dígitos).

```

static int funcaoRA ( int ra ) {
    int i=0, m1, m2;
    int v[] = new int[8];
    while (ra>0) {
        v[i] = (ra%10)+i;
        ra = ra/10;
        i = i+1;
    }
    if (v[0] > v[1]) {
        m1 = v[0];
        m2 = v[1];
    }
    else {
        m2 = v[0];
        m1 = v[1];
    }
    for (i=2; i<8; i=i+2) {
        if (v[i]>v[i+1]) {
            if (v[i]>m1) {
                m1 = v[i];
            }
            if (v[i+1]<m2) {
                m2 = v[i+1];
            }
        }
        else {
            if (v[i]<m2) {
                m2 = v[i];
            }
            if (v[i+1]>m1) {
                m1 = v[i+1];
            }
        }
    }
    return m1-m2;
}

```

RA =  |  |  |  |  |  |  |

v =  |  |  |  |  |  |  |

m1 = ..... // m1 terá o maior valor do vetor v

m2 = ..... // m2 terá o menor valor do vetor v

Resposta: ..... // a diferença entre o maior e o menor valor de v

Para RA=12345678 a resposta será 0

Para RA=87654321 a resposta será 14

**Questão 3 (8 pontos)**

Crie uma função que imprima os elementos **pares** da primeira coluna seguidos dos elementos **pares** da última coluna. A matriz de números inteiros, dada como entrada, pode ser quadrada ou retangular.

**Limitação 1:** Use no máximo 1 laço (pode usar várias condicionais).

**Limitação 2:** Não pode usar vetores auxiliares.

**Formato:** A impressão dos elementos deve ser em apenas 2 linhas (veja exemplos).

Entrada	Saída	Entrada	Saída
<pre> 1  2  6 8  6  9 7 10 12 </pre>	<pre> 8 6 12 </pre>	<pre> 8  2  3  4 5  6  7  8 9 10 11 12 13 14 15 15 </pre>	<pre> 8 4 8 12 </pre>

```

static void imprimirColunas ( int M[][][] ) {
    int i, n = M.length, m = M[0].length;

    for (i=0; i<2*n; i=i+1) {
        if (i<n && M[i][0]%2==0)
            System.out.print(M[i][0] + " ");

        if (i==n)
            System.out.println("");

        if (i>=n && M[i-n][m-1]%2==0)
            System.out.printf(M[i-n][m-1] + " ");
    }
}

```

**Questão 4 (6 pontos)**

Crie uma função que imprima os elementos de uma matriz quadrada, de números inteiros, seguindo a ordem da somatória de seus índices, isto é, seguindo a diagonal secundária. A matriz deve ser percorrida apenas uma única vez.

**Limitação:** Use no máximo 2 laços (pode usar várias condicionais).

**Formato:** A impressão dos elementos deve ser em apenas 1 linha (veja exemplo).

Entrada	Saída
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16	1 2 5 3 6 9 4 7 10 13 8 11 14 12 15 16

```
static void imprimirElementos ( int M[][] ) {  
    int i, j, n = M.length, m = M[0].length;  
  
    for(i=0; i< n+m-1; i++) {  
        for(j=0; j<n; j++) {  
            if(i-j>=0 && (i-j)<m)  
                System.out.print(matriz[j][i-j] + " ");  
        }  
    }  
}
```

**Questão 5** (8 pontos)

Dado um número n, seja inv(n) o número que se obtém invertendo-se a ordem dos dígitos de n. Por exemplo inv(322) = 223. Um número é palíndromo se inv(n)=n. Por exemplo, 34543, 1, 99 são números palíndromos.

Escreva uma função que receba como parâmetro apenas um número inteiro n e verifique se n é palíndromo (devolver true se é palíndromo, caso contrário devolver false).

**Limitação 1:** Não podem ser utilizadas funções de conversão de número a string e vice-versa.

**Limitação 2:** Não pode utilizar nenhum tipo de vetor.

**Limitação 3:** Não pode utilizar nenhuma função matemática especial do Java.

```
static boolean palindromo ( int n ) {  
    int aux, inv=0;  
  
    for (aux=n; aux>0; aux=aux/10)  
        inv = inv*10 + aux%10;  
  
    return n==inv;  
}
```

**Questão Bônus (4 pontos)**

Dados o número real  $x$  e um erro (epsilon) a seguinte recorrência descreve uma função  $F$  que permite aproximar  $e^x$  a um valor  $y$  tal que  $|y - e^x| \leq \text{epsilon}$

$$F(x, \text{epsilon}) = \begin{cases} \frac{1}{F(-x, \text{epsilon})} & , \text{ se } x < 0 \\ (F(\frac{x}{2}, \text{epsilon}))^2 & , \text{ se } x > \text{epsilon} \\ 1 + x & , \text{ se } 0 < x \leq \text{epsilon} \end{cases}$$

Crie uma função recursiva que permita calcular  $e^x$  dado um número real  $x$  e um número real  $\text{epsilon}$

```
static double F ( double x, double epsilon) {
    double t;

    if (x<0)
        return 1/F(-x, epsilon);
    if (x>epsilon) {
        t = F(x/2, epsilon);
        return t*t;
    }
    else
        return 1+x;
}
```

Nome:	RA:
-------	-----

**Instruções para a prova (leia antes de começar):**

- 1) A prova tem a duração de 1h50min.
- 2) A compreensão e interpretação do enunciado é parte integrante da avaliação. A prova é individual e sem consultas.

**Questão 1:** Selecione a opção correta. Resposta correta +2, incorreta -1 (10 pontos)

<pre>static boolean F2 (int p, int q) {     if (true)         return false;     if (p==q)         return true;     else         return false; }</pre>	<p>Para dois números inteiros p e q, a função devolve:</p> <p>(a) true se p é igual a q.  (b) false se p é igual a q.  (c) true independente dos valores de p e q.  <b>(d) false independente dos valores de p e q .</b>  (e) Nenhuma das anteriores.</p>
<pre>static int numeros2 (int x, int y, int z) {     if (x&gt;y) {         if (y&gt;z)             return z;         else             return y;     }     else {         if (x&gt;z)             return z;         else             return x;     } }</pre>	<p>Para três números inteiros diferentes x, y, e z, a função devolve:</p> <p><b>(a) O menor valor entre x, y e z.</b>  (b) O maior valor entre x, y e z.  (c) z e y.  (d) z e x .  (e) Nenhuma das anteriores.</p>
<pre>static void G2 (int n) {     int i;     for (i=1; i&lt;=n*2; i=i+2)         System.out.println("SPAM"); }</pre>	<p>Para um número inteiro positivo n, a função:</p> <p><b>(a) Imprime n vezes a palavra “SPAM”.</b>  (b) Imprime n-1 vezes a palavra “SPAM”.  (c) Imprime 2n vezes a palavra “SPAM”.  (d) Imprime 2n-1 vezes a palavra “SPAM”.  (e) Nenhuma das anteriores.</p>
<pre>static int H( int v[ ] ) {     int i;     int n = v.length;     int soma = 0;      for (i=0; i&lt;n; i=i+1)         soma = 2*(soma + v[i]);      return soma; }</pre>	<p>Para v={4,3,1,2}, a função devolve</p> <p>(a) 80.  (b) 86.  (c) 90.  <b>(d) 96.</b>  (e) Nenhuma das anteriores.</p>
<pre>static String J( String p ) {     int i;     String r = "";      for (i=0; i&lt;p.length(); i=i+2) {         r = r + p.charAt(i);         r = r + p.charAt(i) + r;     }      return r; }</pre>	<p>Para p="bca", a função devolve</p> <p>(a) "aaacccaaac".  (b) "aaabbaaaab".  <b>(c) "bbaabbba".</b>  (d) "bbbccbbbc".  (e) Nenhuma das anteriores.</p>

**Questão 2:** (8 pontos)

Indique o valor que devolverá o seguinte método. Considere como parâmetro de entrada, para o método, o seu número de matrícula (isto é, seu RA composto de 8 dígitos).

```

static int funcaoRA ( int ra ) {
    int i=0, m1, m2;
    int v[] = new int[8];
    while (ra>0) {
        v[i] = (ra%10)+i;
        ra = ra/10;
        i = i+1;
    }
    if (v[0] > v[1]) {
        m1 = v[0];
        m2 = v[1];
    }
    else {
        m2 = v[0];
        m1 = v[1];
    }
    for (i=2; i<8; i=i+2) {
        if (v[i]>v[i+1]) {
            if (v[i]>m1) {
                m1 = v[i];
            }
            if (v[i+1]<m2) {
                m2 = v[i+1];
            }
        }
        else {
            if (v[i]<m2) {
                m2 = v[i];
            }
            if (v[i+1]>m1) {
                m1 = v[i+1];
            }
        }
    }
    return m1-m2;
}

```

RA =  |  |  |  |  |  |  |

v =  |  |  |  |  |  |  |

m1 = ..... // m1 terá o maior valor do vetor v

m2 = ..... // m2 terá o menor valor do vetor v

Resposta: ..... // a diferença entre o maior e o menor valor de v

Para RA=12345678 a resposta será 0

Para RA=87654321 a resposta será 14

**Questão 3 (8 pontos)**

Crie uma função que imprima os elementos **ímpares** da primeira coluna seguidos dos elementos **ímpares** da última coluna. A matriz de números inteiros, dada como entrada, pode ser quadrada ou retangular.

**Limitação 1:** Use no máximo 1 laço (pode usar várias condicionais).

**Limitação 2:** Não pode usar vetores auxiliares.

**Formato:** A impressão dos elementos deve ser em apenas 2 linhas (veja exemplos).

Entrada	Saída	Entrada	Saída
<pre> 1  2  6 8  6  9 7 10 12 </pre>	<pre> 1 7 9 </pre>	<pre> 8  2  3  4 5  6  7  8 9 10 11 12 13 14 15 15 </pre>	<pre> 5 9 13 15 </pre>

```

static void imprimirColunas ( int M[][][] ) {
    int i, n = M.length, m = M[0].length;

    for (i=0; i<2*n; i=i+1) {
        if (i<n && M[i][0]%2==1)
            System.out.print(M[i][0] + " ");

        if (i==n)
            System.out.println("");

        if (i>=n && M[i-n][m-1]%2==1)
            System.out.printf(M[i-n][m-1] + " ");
    }
}

```

**Questão 4** (6 pontos)

Crie uma função que imprima os elementos de uma matriz quadrada, de números inteiros, seguindo a ordem da somatória de seus índices, isto é, seguindo a diagonal secundária. A matriz deve ser percorrida apenas uma única vez.

**Limitação:** Use no máximo 2 laços (pode usar várias condicionais).

**Formato:** A impressão dos elementos deve ser em apenas 1 linha (veja exemplo).

Entrada	Saída
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16	1 2 5 3 6 9 4 7 10 13 8 11 14 12 15 16

```
static void imprimirElementos ( int M[][] ) {  
    int i, j, n = M.length, m = M[0].length;  
  
    for(i=0; i< n+m-1; i++) {  
        for(j=0; j<n; j++) {  
            if(i-j>=0 && (i-j)<m)  
                System.out.print(matriz[j][i-j] + " ");  
        }  
    }  
}
```

**Questão 5** (8 pontos)

Dado um número n, seja inv(n) o número que se obtém invertendo-se a ordem dos dígitos de n. Por exemplo inv(322) = 223. Um número é palíndromo se inv(n)=n. Por exemplo, 34543, 1, 99 são números palíndromos.

Escreva uma função que receba como parâmetro apenas um número inteiro n e verifique se n é palíndromo (devolver true se é palíndromo, caso contrário devolver false).

**Limitação 1:** Não podem ser utilizadas funções de conversão de número a string e vice-versa.

**Limitação 2:** Não pode utilizar nenhum tipo de vetor.

**Limitação 3:** Não pode utilizar nenhuma função matemática especial do Java.

```
static boolean palindromo ( int n ) {  
    int aux, inv=0;  
  
    for (aux=n; aux>0; aux=aux/10)  
        inv = inv*10 + aux%10;  
  
    return n==inv;  
}
```

**Questão Bônus (4 pontos)**

Dados o número real  $x$  e um erro (epsilon) a seguinte recorrência descreve uma função  $F$  que permite aproximar  $e^x$  a um valor  $y$  tal que  $|y - e^x| \leq \text{epsilon}$

$$F(x, \text{epsilon}) = \begin{cases} \frac{1}{F(-x, \text{epsilon})} & , \text{ se } x < 0 \\ (F(\frac{x}{2}, \text{epsilon}))^2 & , \text{ se } x > \text{epsilon} \\ 1 + x & , \text{ se } 0 < x \leq \text{epsilon} \end{cases}$$

Crie uma função recursiva que permita calcular  $e^x$  dado um número real  $x$  e um número real  $\text{epsilon}$

```
static double F ( double x, double epsilon) {
    double t;

    if (x<0)
        return 1/F(-x, epsilon);
    if (x>epsilon) {
        t = F(x/2, epsilon);
        return t*t;
    }
    else
        return 1+x;
}
```