PROGRAMACIÓN JAVA. SOLUCIONES EJERCICIOS TEMA 5. Trazas - ARRAYS y STRINGS.

Para cada uno de los siguientes bloques de instrucciones, explica razonadamente lo que imprimen o en su caso si producen algún tipo de error:

```
1. int [] v = {10, 20, 30, 40, 50};
   int x, suma = 0;
   for(x=4; x >= 0; x--)
       suma = suma + v[x] + x;
   System.out.println(suma);
```

```
 x     suma     v[x]              v                                  Salida
-------------------------------------------------------------------------
 4       0      v[4]-> 50         10, 20, 30, 40, 50                   160
 3      54      v[3]-> 40
 2      97      v[2]-> 30
 1     129      v[1]-> 20
 0     150      v[0]-> 10
-1     160
```

```
2. int[][] m1 = {{1, 2, 3}, {4, 5, 6}, {7, 8, 9}};
   int N = m1.length;
   int[][] m2 = new int[N][N];
   int x, y;
   for (x = 0; x < N; x+=2) {
       for (y = 0; y < N; y++) {
            if (m1[x][y] % 2 == 0) {
                m2[x][y] = 0;
            } else {
                m2[x][y] = 1;
            }
       }
    }
    System.out.println(m2[0][1] + " " + m2[1][1] + " " + m2[2][1]);
```

```
  m1    N      m2     x    y   m1[x][y]          m2[x][y]
-------------------------------------------------------------------------------
1 2 3   3    0 0 0    0    0   m1[0][0]-> 1      m2[0][0]=1
4 5 6        0 0 0
7 8 9        0 0 0

             1 0 0
             0 0 0
             0 0 0

                           1   m1[0][1]-> 2      m2[0][1]= 0 (mismo valor que tiene ahora)
                           2   m1[0][2]-> 3      m2[0][2]= 1
             1 0 1
             0 0 0
             0 0 0
                           3
                     2     0   m1[2][0]-> 7      m2[2][0]= 1
             1 0 1
             0 0 0
             1 0 0
                           1   m1[2][1]-> 0      m2[2][1]= 0 (mismo valor que tiene ahora)
                           2   m1[2][2]-> 9      m2[2][2]= 1
             1 0 1
             0 0 0
             1 0 1
                           3
                     4
 ---> salida: 0 0 0
```

```
3. String str1 = "Estamos aprendiendo Java ahora";
   String str2 = "Estamos APRENDIENDO Java";
   String s;
   if (str1.compareToIgnoreCase(str2) > 0){
       s = " es mayor que ";
   }else if (str1.compareTo(str2) < 0){
            s = " es menor que ";
   }else{
            s = " es igual a ";
   }
   System.out.println(str1 + s + str2);
```

El método compareToIgnoreCase compara alafbéticamente dos String.

La instrucción:

```
if (str1.compareToIgnoreCase(str2) > 0)
```

compara si str1 es mayor que str2 en orden alfabético

En este caso es cierto.

La cadena "Estamos aprendiendo Java ahora" alfabéticamente es mayor que la cadena "Estamos APRENDIENDO Java". Esto quiere decir que en un listado alfabético primero aparecería "Estamos APRENDIENDO Java" y después "Estamos aprendiendo Java ahora".

Como la condición se cumple, se realiza la asignación: s = " es menor que ";

Se muestra por pantalla:

```
Estamos aprendiendo Java ahora es mayor que Estamos APRENDIENDO Java
```

```
4. int[][] m = {{1, 2, 3, 4}, {5, 6, 7}, {8, 9, 0, 1}};
   int i, j = 1;
   for (i = 1; i < m.length; i++) {
        switch(m[i][j]){
               case 3:
               case 6:  m[i][j]=m[i][j]+m[i-1][j];
               default: m[i-1][j]++;
         }
   }
   System.out.println(m[0][1] + " " + m[1][1] + " " + m[2][1]);
```

```
     m           m.length       i    j
-------------------------------------------------------------------------------------------
1  2  3  4          3
5  6  7
8  9  0  1
                                1    1    switch(m[i][j])-> switch(m[1][1])-> 6
                                          case 6: m[i][j] = m[i][j]+m[i-1][j];
                                                  m[1][1] = m[1][1]+m[0][1];
                                                  m[1][1] =   6   +    2
                                                  m[1][1] = 8
1  2  3  4
5  8  7
8  9  0  1                                    (También se ejecuta el default porque no hay break)
                                           default: m[i-1][j]++;
                                                    m[0][1]++;
1  3  3  4
5  8  7
8  9  0  1
                                2          switch(m[i][j])-> switch(m[2][1])-> 9
                                           default: m[i-1][j]++;
                                                    m[1][1]++;
1  3  3  4
5  9  7
8  9  0  1
                                3

---> salida: 3 9 9
```

```
5.  int[] a = {0, 2, 3, 1};
    int[] b = {0, 4, 2, 1, 3};
    int x = 0;
    switch (b[a.length]) {
           case 1:
           case 2: x--;
           case 3: x--;
           case 0: x--;
           default: x--;
           break;
    }
    System.out.println(x);
```

a	b	a.length	b[a.length]		x	salida
0 2 3 1	0 4 2 1 3	4			0	-3
			b[4]-> 3	case 3: x--;	-1	
				x--;	-2	
				x--;	-3	

El switch entra en el case 3 y se ejecuta hasta encontrar el break;

```
6.  String[] s = {"uno", "dos", "tres", "cuatro"};
    for (int i = s.length - 1; i > 0; i--) {
         System.out.println(s[i].charAt(i));

    }
```

s	s.length	i	s[i]	s[i].charAt(i)	salida
{"uno", "dos", "tres", "cuatro"}	4	3	"cuatro"	t	t
		2	"tres"	e	e
		1	"dos"	o	o
		0			

```
7. int[][] m = {{1, 2, 3}, {4, 5}, {6, 7, 8, 9}};
    int i, j = 0;
    for (i = 1; i < m.length; i++) {
        m[i][j] = m[i][j] + m[i - 1][j];
    }
    System.out.println(m[0][0] + " " + m[1][0] + " " + m[2][0]);
    System.out.println(m[0][1] + " " + m[1][1] + " " + m[2][1]);
```

m	m.length	i	j	m[i][j]	m[i-1][j]	m[i][j] = m[i][j] + m[i - 1][j];
1 2 3 4 5 6 7 8 9	3	1	0	4	1	5
1 2 3 5 5 6 7 8 9		2		6	5	11
1 2 3 5 5 11 7 8 9		3				

---> salida: 1 5 11
2 5 7

8.
```
int[] a = {0, 2, 3, 1};
int[] b = {11, 22, 33, 44};
int i = 3;
do {
    System.out.println(b[a[i]] + i);
} while (--i > 1);
```

a	b	i	a[i]	b[a[i]]	salida
0 2 3 1	11 22 33 44	3	1	22	25
		2	3	44	46
		1			

9.
```
String s1 = "123";
String s2 = "45";
String s3 = ".";
String s = s1 + s2 + s3;
for (int i = 1; i < s.length(); i+= 2) {
     System.out.print(s.charAt(i));
}
```

s1	s2	s3	s	s.length()	i	s.charAt(i)	salida
"123"	"45"	"."	"12345."	6	1	2	24.
					3	4	
					5	.	
					7		

10.
```
public static void main(String[] args) {
        int b = 1;
        int[] A = {0, 1, 2, 0};
        A[3] = ++b;
        A[0]++;
        metodo(A, b);
        System.out.println(A[0] + " " + A[1] + " " + A[2] + " " + A[3]);
    }

    public static void metodo(int[] X, int n) {
        int k = n / X[2];
        switch (k) {
            case 0:  k++;
            case 1:
            case 2:  n *= 2;
                     break;
            default: n = 0;
        }
        X[1] = k + n;
    }
```

main					metodo			
A	b	A[3]	A[0]	salida	X	n	k	X[1]
0 1 2 0	1							
	2	2						
0 1 2 2			1					
1 1 2 2					1 1 2 2	2	1	
						4		5
					1 5 2 2			
1 5 2 2				1 5 2 2				

```
11. String s = "abc-abc-abc";
    int k = s.indexOf("a", 1);
    int x = s.lastIndexOf("-");
    s = s.substring(k,x);
    for (int i = 1; i < s.length(); i++) {
        System.out.print(s.charAt(i) + "*");
    }
```

```
      s           k      x    s.length i       salida
-------------------------------------------------------------------------------------------------
"abc-abc-abc"     4      7
"abc"                            3
                                      1         b*c*
                                      2
                                      3
```

```
12. String s1 = "Uno";
    String s2 = s1 + "Dos";
    String s3 = s1 + s2 + "Tres";
    int a = s3.lastIndexOf(s1.charAt(1));
    int b = s3.indexOf("o", 3);
    int c = s2.length();
    int d = s1.length() + s2.length();
    String s4 = s3.substring(a, c);
    String s5 = s3.substring(b, d);
    System.out.println(s4.toUpperCase() + "-" + s5.toLowerCase() + "-" + s3.charAt(b+1));
```

```
 s1        s2           s3           a   b   c   d     s4      s5          salida
-------------------------------------------------------------------------------------------------
"Uno"   "UnoDos"  "UnoUnoDosTres"    4   5   6   9    "no"   "oDos"        NO-odos-D
```

```
13. public static void main(String[] args) {
        int[][] m = {{1, 2, 3, 4}, {4, 5}, {8, 9, 6}, {0, 1, 2, 3, 4}};
        metodo(m);
        for (int i = 0; i < m.length; i++) {
            System.out.print(m[i][1] + " ");
        }
    }

    public static void metodo(int[][] y) {
        for (int i = 1, j = 1; i < y[0].length; i++) {
            switch (y[i][j]) {
                case 3:
                case 5:
                case 9:   y[i][j] = y[i][j] + y[i - 1][j];
                default:  y[i - 1][j]++;
            }
        }
    }
```

main

m	m.length	i	salida
1 2 3 4 4 5 8 9 6 0 1 2 3 4			
1 3 3 4 4 8 8 16 6 0 1 2 3 4	4	0	3 8 17 1
		1	
		2	
		3	
		4	

metodo

y	y[0].length	i	j	switch(y[i][j])
1 2 3 4 4 5 8 9 6 0 1 2 3 4	4	1	1	5->case 5 hasta fin switch y[1][1] = 5 + 2 = 7 y[0][1] = 3
1 3 3 4 4 7 8 9 6 0 1 2 3 4		2		9->case 9 hasta fin switch y[2][1] = 9 + 7 = 16 y[1][1] = 8
1 3 3 4 4 8 8 16 6 0 1 2 3 4		3		1->default y[2][1] = 17
1 3 3 4 4 8 8 17 6 0 1 2 3 4		4		

```
14. public static void main(String[] args) {
      int[] a = {1, 2, 1, 3, 4};
      int[][] m = {{1, 2, 3}, {4, 5, 6}, {7,8,9}};
      metodo(a);
      int i = 1, j = 1;
      do {
         if(a[i]%2==0){
            m[i][j] = m[i-1][j]++;
         }
      } while (i++<3);
      for (int x : a) {
           System.out.println(x);
      }
      System.out.println(m[0][1]+ "" + m[1][1] + "" + m[2][1]);
   }

   public static void metodo(int [] x){
          for(int i = 1; i < x.length-2; i++){
              x[i] += x[i+1];
          }

   }
```

```
                main                                    metodo
----------------------------------------  ------------------
   a          m    i  j  x     salida         x        i
----------------------------------------  ------------------
1 2 1 3 4    1 2 3                         1 2 1 3 4    1
             4 5 6
             7 8 9                         1 3 1 3 4    2

                                           1 3 4 3 4    3

1 3 4 3 4           1  1
                    2
             1 2 3
             4 6 6
             7 5 9
                    3
                    4     1       1
                          3       3
                          4       4
                          3       3
                          4       4
                                  265
```

```
15. String s1 = "abcd";
    String s2 = s1 + "efgh";
    String s3 = s1.substring(1,3) + s2.substring(2,5);
    int x = s3.lastIndexOf("f");
    int y = s3.indexOf("c", 1);
    for(int i = x+1; i <=2; i++){
        System.out.print(s3.charAt(i) + ", " + s3.charAt(i+1));
    }
```

```
  s1        s2         s3       x   y   i    salida
-------------------------------------------------------------------------------------------
"abcd"  "abcdefgh"  "bccde"   -1   2   0    b, cc, cc, d
                                       1
                                       2
                                       3
```

16.
```
int[] A = {1, 2, 3, 4};
A[0] += A[1];
A[2] = A[0] + A[3];
metodo(A);
System.out.println(A[0] + " " + A[1] + " " + A[2] + " " + A[3]);
public static void metodo(int[] X) {
        int k = X[2]++;
        switch (k) {
            case 5:  k++;
            case 8:
            case 4:  k *= 2;
            default: k = 1;
        }
        X[3] = k;
 }
```

main				metodo		
A	A[0]	A[2]	salida	X	k	switch(k)
1 2 3 4	3					
3 2 3 4		7				
3 2 7 4				3 2 7 4	7	
				3 2 8 4	1	switch(7) -> default
				3 2 8 1		
3 2 8 1			3 2 8 1			

17.
```
String s = "abc8defg8hijklmn";
int k = s.lastIndexOf("8");
for (int i = 1; i < k; i += 4) {
     System.out.print(s.substring(i, i + 3) + "-");
}
```

s	k	i	salida
"abc8defg8hijklmn"	8	1	bc8-efg-
		5	
		9	

18.
```
double[][] m = {{1.1, 2.2, 3.3}, {4.4, 5.5}, {6.6, 7.7, 8.8}};
int i, j = 1, k = 1;
for (i = 1; i < 3; i++) {
      switch(m[i][j]){
            case 4: k++;
                    break;
            case 4.4: k++;
            default: k++;
      }
}
System.out.println(k);
```

Error: switch no admite el tipo double en la expresión a comprobar ni en los case.