

13.75

~~13.75~~

Ex. 1

① S / s  
init: @ 12.4

	i	u	v	x	y	output on screen
0	0	0	0	0	0	fin: 0
1	0	1	0	0	0	v=1 fin: 0
2	0	12	0	0	0	v=12 fin: 0
3	1	0	0	0	0	<del>fin: 0</del> fin: 0     1,5
4	1	4	4	0	0	v=4 fin: 4

② v shows the last integer in the string and fin shows the last signed integer (-% for plus) calculates the sum of all the signed integers from the string starting from the second if the first character in string is not alpha-numeric a digit, or a sign for the following integer.

1.75



Ex. 2

```

public static double computeSquareRoot (int a, double ERROR) {
    final double ERROR = 0.000001; int k;
    int u = 1;
    if (a >= 0) {
        do {
            k = u;
            u = 0.5 * (u + a / u);
        } while ((u >= k - ERROR) && (u <= k + ERROR));
    } else {
        u = -u;
        System.out.println ("ERROR!");
    }
    return u;
}

```

!!!

2,5

Ex. 3

```

Ex. 1 public static int inputBetweenBounds (int bMin, int bMax) {
    int x;
    System.out.println ("How many tokens?");
    int x = readInteger();
    if ((x > bMax) || (x < bMin)) {
    if (x > bMax) {
        & System.out.println ("Enter a lower number, please!");
        x = read
    }
    do
    if ((x > bMax) || (x < bMin)) {
        do {
            if (x > bMax) {
                System.out.println ("Enter a lower number!");
                x = readInteger();
            }
        }
    }
}

```

1,5



```

    } else {
        System.out.println("Enter a bigger number!");
        x = readInteger();
    }
} while ((x > bMax) || (x < bMin))
}
return x;
}

```

Q2

```

public static void displayArray (int [] t) {
    for (int i=0; i < t.length; i++) {
        System.out.print (t[i] + " ");
    } System.out.println();
}

```

**Q, 1**

Q3

```

public static int [] generateArrayInt (int k, int nbMax) {
    double k,
    int [] a = new int [k];
    for (int i=0; i < a.length; i++) {
        a[i] = Math.random() * 10;
        a[i] = (int)k nbMax % (int)k;
    }
    return a;
}

```

**1**



```

24 public static boolean over (int [] tab) {
    boolean k = false; k = true;
    for (int i = 0; i < tab.length; i++) {
        if (tab[i] == 0) {
            k = false;
            break;
        }
    }
    return k;
}

```

NO it is true if ALL are empty!  
0,5

```

25 public class Min {
    public static main (String [] args) {
        int MAX int i = 0;
        final int MAX = 20;
        int k = 0;
        final int N = 13;
        int z = 0;
        int [] a = new int [N];
        a = generateArrayInt (N, MAX);
        do {
            displayArray (int a);
            k = inputBetweenBounds (1, System.out.println ("Select the loop"));
            k = readInteger ();
            z = inputBetweenBounds (1, a[k]);
            a[k] = a[k] - z;
            i++;
        } while (over (a) != true);
        System.out.println (i % 2); i
        if (i % 2 == 0) {
            i = 2;
        } else {
            i = 1;
        }
        System.out.println (i);
    }
}

```

MORE DISPLAY  
TO FOLLOW  
THE GAME

← if a[k] is empty???

OK



Q6

```

public static int toDecimal (int[] tab) {
    int n = tab.length, tab.length;
    int k = 0;
    for (int i = tab.length - 1, i >= 0, i--) {
        k = k + tab[i] * (int) Math.pow(2, i);
        // n--;
    }
    return k;
}

```

1

Q7

```

public static int[] sumAin (int[] bi, int[] bj) {
    int[] c = new int[4];
    for (int i = bi.length; i >= 0, i--) {
        c[i] = bi[i] + bj[i];
        if (c[i] == 2) {
            c[i] = 0;
            c[i-1] = 1;
        } else if (c[i] == 3) {
            c[i] = 1;
            c[i-1] = 1;
        }
    }
    c[0] = bi[0] + bj[0];
    if (c[0] == 2)
        c[0] = 0;
    if (c[0] == 3)
        c[0] = 1;
    return c;
}

```

0,75

it is a XOR !!

