

Prénom :

Nom :

Pour chaque programme, vous devez indiquer l'affichage produit en répondant sur l'énoncé, à droite du programme.
Soyez très précis dans vos réponses.

```

3 public class Exo11 {
4
5     public static void main(String[] args) {
6         int k = 2;
7         for (int j = 1; j < 5; j++) {
8             k = k + 1;
9             System.out.printf("%d %d\n", j, k);
10            k = k - j;
11        }
12        System.out.println(k);
13        boolean b = true;
14        int x = 20;
15        while (x > 1) {
16            b = b && (x > 4);
17            System.out.printf("%b\n", b);
18            x = x/2;
19            if(!b) {
20                b = !b;
21            }
22            System.out.printf("%d\n", x);
23        }
24    }
25 }

```

Affichage produit

```

3 import java.util.Arrays;
4
5 public class Exo21 {
6
7     public static void main(String[] args) {
8         int[] toto = { 12, 3, 4, 7, -4 };
9         System.out.printf("%s\n", Arrays.toString(toto));
10        System.out.printf("%s\n", toto);
11        int j = 0;
12        for (int x : toto) {
13            System.out.printf("[%d] %d, ", x, j);
14            j = j + x;
15        }
16        System.out.println(j);
17        int[] foo = new int[toto.length - 1];
18        System.out.printf("%s\n", Arrays.toString(foo));
19        int k = 1;
20        while (k - 1 < foo.length) {
21            foo[k - 1] = toto[toto.length - k] + k;
22            k++;
23        }
24        System.out.printf("%s\n", Arrays.toString(foo));
25    }
26 }
27 }

```

Affichage produit

```

3 import java.util.Arrays;
4
5 public class Exo31 {
6
7     public static void main(String[] args) {
8         int[] a = { 1, 2, 3 };
9         int[] b = a;
10        int[] c = new int[a.length];
11        for (int i = 0; i < a.length; i++) {
12            c[i] = a[i] + 1;
13            b[i] = a[i] - 1;
14        }
15        System.out.printf("%s%n", Arrays.toString(a));
16        System.out.printf("%s%n", Arrays.toString(b));
17        System.out.printf("%s%n", Arrays.toString(c));
18    }
19
20 }

```

Programmation (1)

Écrire un programme **complet** qui réalise les opérations suivantes :

1. demande à l'utilisateur une valeur entière k en s'assurant que $k > 0$ (on redemande la valeur tant qu'elle n'est pas acceptable) ;
2. demande à l'utilisateur une valeur réelle y ;
3. affiche dans l'ordre les valeurs de la suite $(u_j)_{j \geq 1}$ (définie ci-dessous) pour j allant de 1 à k inclus, sous la forme (ici avec $k = 3$ et $y = 1.0$) :

```

u_1 = 0.0
u_2 = 1.0
u_3 = 2.0

```

La suite $(u_j)_{j \geq 1}$ est donnée par :

$$\begin{aligned}
 u_1 &= 0 \\
 u_{2m} &= m + u_{2m-1} \times y && \text{pour tout } m \geq 1 \\
 u_{2m+1} &= u_{2m} - u_{2m-1} + y && \text{pour tout } m \geq 1
 \end{aligned}$$

Programmation (2)

Écrire un programme (réduit au contenu de la méthode `main`) qui réalise les opérations suivantes :

1. crée un tableau `tab` de longueur 3 ;
2. demande à l'utilisateur trois chiffres (donc entre 0 et 9 inclus) à placer dans le tableau ;
3. demande à l'utilisateur un entier k ;
4. construit un tableau `tab2` de longueur $3k$ contenant k copies du tableau `tab` ;
5. ajoute à chaque case i du tableau `tab2` la valeur i (donc 0 à la première, 1 à la deuxième, etc.) ;
6. affiche le tableau `tab2` résultat sous la forme indiquée ci-dessous.

Si l'utilisateur entre les chiffres 1, 2, 3, dans cet ordre, puis donne $k = 3$, on doit donc obtenir un tableau à 9 cases, contenant d'abord 1, 2, 3, 1, 2, 3, 1, 2, et 3, après l'étape 4, puis 1, 3, 5, 4, 6, 8, 7, 9, et 11 après l'étape 5. On l'affichera sous la forme

```

{ _1_3_5_ }
{ _4_6_8_ }
{ _7_9_11_ }

```

Attention, l'affichage doit se faire impérativement à partir du tableau final, par en calculant les valeurs à afficher pendant l'affichage lui-même.