

Boucles avancées

Exercice 3.2.1

Donner la trace de l'exécution du programme suivant. Qu'affiche-t-il ?

```
public class Prog1 {  
    public static void main(String args[]) {  
        for (int i= 1; i <= 2; i++) {  
            for (int j= 1; j <= i; j++) {  
                int k= i*j;  
                if (k < 10)  
                    Terminal.ecrireChar('0');  
                Terminal.ecrireString(k + " ");  
            }  
            Terminal.sautDeLigne();  
        }  
    }  
}
```

	i	j	k	$i \leq 3$	$j \leq i$	$k \leq 10$	affichage
3 (init)	1	X	X				
3 (test)	1	X	X	V			
4 (init)	1	1	X				
4 (test)	1	1	X		V		
5	1	1	1				
6	1	1	1			V	
7	1	1	1				0
8	1	1	1				1_
4 (incrément)	1	2	X				
4 (test)	1	2	X		F		
10	1	X	X				(saut de ligne)
3 (incrément)	2	X	X				
3 (test)	2	X	X	V			
4 (init)	2	1	X				
4 (test)	2	1	X		V		
5	2	1	2				
6	2	1	2			V	
7	2	1	2				0
8	2	1	2				2_
4 (incrément)	2	2	X				
4 (test)	2	2	X		V		
5	2	2	4				
6	2	2	4			V	
7	2	2	4				0
8	2	2	4				4_
4 (incrément)	2	3	X				
4 (test)	2	3	X		F		
10	2	X	X				(saut de ligne)
3 (incrément)	3	X	X				
3 (test)	3	X	X	F			

L'affichage est donc :

```
01
02 04
```

Exercice 3.2.2

Question 1

Ecrire un programme qui affiche un carré composé de caractères ' * ', dont la taille est entrée par l'utilisateur.

Par exemple, pour une taille 4, ce sera :

```
****
****
****
```

Il y a plusieurs façons de réaliser cela. Une d'elle consiste à utiliser une seule boucle, une autre utilise deux boucles imbriquées l'une dans l'autre. Vous privilégiez la méthode qui utilise deux boucles.

```
public class Corr5_2_1 {
    public static void main(String args[] {
        Terminal.ecrireString("taille_du_carré:");
        int taille= Terminal.lireInt();

        for (int i= 0; i < taille; i++) {
            for (int j= 0; j < taille; j++) {
                Terminal.ecrireChar('*');
            }
            Terminal.sautDeLigne();
        }
    }
}
```

Question 2 *triangle*

Dans la lignée de l'exercice précédent, il s'agit de dessiner cette fois un triangle avec la pointe tournée vers le haut.

Pour le triangle, on demandera à l'utilisateur de rentrer le nombre de lignes. Pour chaque ligne, il faudra afficher un certain nombre d'espaces, puis un certain nombre d'étoiles, puis un `New_Line`. Vous devrez mettre en évidence une relation entre le numéro de ligne , la taille et le nombre d'espace et une autre relation entre le numéro de ligne , la taille et le nombre d'étoiles.

```
 *
***
*****
*****
```

```
public class Corr5_2_2 {
    public static void main(String args[] {
        Terminal.ecrireString("hauteur_du_triangle:");
        int taille= Terminal.lireInt();

        // Nombre d'espaces à dessiner avant les étoiles
        int nEspaces= taille -1;
        // Nombre d'étoiles à dessiner
        int nEtoiles= 1;

        // Pour chaque ligne...
        for (int i= 0; i < taille; i++) {
            // On dessine les espaces...
            for (int j= 0; j < nEspaces; j++) {
                Terminal.ecrireChar(' ');
            }
            // On dessine les étoiles
```

```

    for (int j= 0; j < nEtoiles; j++) {
        Terminal.ecrireChar('*');
    }
    // On passe à la ligne...
    Terminal.sautDeLigne();
    // On met à jour ...
    nEspaces= nEspaces -1;
    nEtoiles= nEtoiles + 2; // (une de chaque côté)
}
}
}

```

Exercice 3.2.3

Question 1

(boucle simple)

Réalisez un programme qui dira si un nombre saisi au clavier est premier. Pour cela, on verra s'il est divisible par au moins un nombre compris entre 2 et lui-même - 1. Par exemple, 5 est premier car il n'est divisible ni par 2, ni par 3, ni par 4. Remarque : 1 n'est pas considéré comme premier.

Pour savoir si a est divisible par b, il suffit de tester si $a \% b == 0$

```

public class Corr5_3_1 {
    public static void main(String args[]) {
        Terminal.ecrireString("Entrez_un_nombre_");
        int a= Terminal.lireInt();

        // On considère qu'un nombre est premier jusqu'à preuve du contraire...
        boolean premier= true;
        if (a < 2) // on gère le cas de 1.
            premier= false;
        else {
            for (int k= 2; premier && k < a; k++) {
                if (a % k == 0)
                    premier = false;
            }
        }
        if (premier)
            Terminal.ecrireStringln(a + "_est_premier");
        else
            Terminal.ecrireStringln(a + "_n'est_pas_premier");
    }
}

```

Question 2 (boucles imbriquées)

Écrire un programme qui affichera les 100 premiers nombres premiers (attention, on n'a pas dit les nombres premiers inférieurs à 100 !).

Réfléchissez au type de boucle le plus approprié pour la boucle extérieure.

```

public class Corr5_3_2 {
    public static void main(String args[]) {
        int nombreDePremiers= 0;
        int a= 2; // On évite de gérer le cas particulier de 1.

        while (nombreDePremiers < 100) {
            boolean premier= true;
            for (int k= 2; premier && k < a; k++) {
                if (a % k == 0)
                    premier = false;
            }
            if (premier) {
                Terminal.ecrireIntln(a);
                nombreDePremiers++;
            }
            a= a+1;
        }
    }
}

```

Exercice 3.2.4 *trace de programme*

Donner la trace de l'exécution du programme suivant, si l'utilisateur rentre 2 et 5 comme valeurs de a et b. Que calcule ce programme ?

```

public class Puis {
    public static void main(String args[]) {
        int a,b;
        Terminal.ecrireString("entrez_a_");
        a= Terminal.lireInt();
        Terminal.ecrireString("entrez_b_");
        b= Terminal.lireInt();

        int r= 1;
        while (b != 0) {
            if (b % 2 == 0) {
                a= a*a;
                b= b/2;
            } else {
                r= r*a;
                b= b-1;
            }
        }
        Terminal.ecrireStringln("résultat_ " + r);
    }
}

```

	a	b	r	b != a	b % 2 == 0	entrée	affichage
3	?	?	X				
4	?	?	X				entrez a
5	2	?	X			2	
6	2	?	X				entrez b
7	2	5	X			5	
9	2	5	1				
10	2	5	1	V			
11	2	5	1	F			
15	2	5	2				
16	2	4	2				
10	2	4	2	V			
11	2	4	2		V		
12	4	4	2				
13	4	2	2				
10	4	2	2	V			
11	4	2	2		V		
12	16	2	2				
13	16	1	2				
10	16	1	2	V			
11	16	1	2		F		
15	32	1	2				
16	32	0	2				
10	32	0	2	F			
19	32	0	2				résultat 32