

# SERIE 3– SOLUTION

## Informatique 1 | Informatik 1

### Question | Frage 1

Gegeben sei der folgende Code: | Soient les déclarations suivantes:

```
1 int a; int b; int c;
```

- (a) Schreiben Sie Code, mit dem der Höchstwert dreier Variablen *a*, *b* und *c* angezeigt werden kann.  
Écrivez le code permettant d'afficher le maximum des trois variables *a*, *b* et *c* (càd la plus grande des trois variables).

#### Solution:

```
1 int max = a;
2 if(b > max)
3 {
4     max = b;
5 }
6
7 if(c > max)
8 {
9     max = c;
10 }
11
12 System.out.println(max);
```

#### Short form:

```
1 int max = a > b ? a : b;
2 max = c > max ? c : max;
3 System.out.println(max);
```

- (b) Schreiben Sie Code, die Quadratwurzel von *a* angezeigt werden kann. Wenn *a* negativ ist, erscheint eine Fehlermeldung. Verwenden Sie dazu die Funktion `Math.sqrt(double)`, welche die Wurzel eines Werts vom Typ `double` zurückgibt, der als Parameter verwendet wird.

Écrivez le code pour afficher la racine carrée du nombre *a*. Si ce nombre est négatif, affichez un message d'erreur. Pour réaliser cela, utiliser la fonction `Math.sqrt(double)` qui retourne la racine d'un `double` passé en paramètre.

#### Solution:

```
1 if(a < 0)
2 {
3     System.out.println("a is negative !");
4 }
5 else
6 {
7     System.out.println("The square root of a is : " + Math.sqrt(a));
8 }
```

### Question | Frage 2

Gegeben sei der folgende Code:

Soit le programme suivant:

```
1 int x = 0;
2 for(int i = 0; i < 10; i++)
3 {
4     x = i+1;
```

Listing continues on next page →

...continuing from previous page

```
5 System.out.println(x);
6 }
```

(a) Was zeigt den folgenden Programm an? Qu'affiche le programme ci-dessus ?

**Solution:**

```
1
2
3
...
9
10
```

(b) Wenn  $x$  vor der Schleife gleich 22 ist, welchen Wert hat  $x$  nach der Schleife?  
Si  $x$  vaut 22 avant la boucle, quelle est sa valeur après celle-ci ?

**Solution:**  $x = 10$

### Question | Frage 3

Gegeben sei der folgende Code:

Soit la portion de code suivante :

```
1 int x = 0;
2 for(int i = 0; i < 10; x++) {
3     System.out.println("Number : " + x);
4 }
```

(a) Ist dieser Code kompilierbar? Begründen Sie Ihre Antwort  
Peut-on compiler ce code ? Justifiez votre réponse?

**Solution:** Ja, weil die Syntax korrekt ist.  
Oui, car il est syntaxiquement correct.

(b) Welches Resultat wird durch die Ausführung dieses Codes ausgegeben?  
Quel est le résultat de l'exécution de ce code ?

**Solution:**

```
Number : 0
Number : 1
Number : 2
[...]
```

(c) Welches Problem entsteht durch die Ausführung dieses Codes?  
Quel est le problème de l'exécution de ce code ?

**Solution:** Weil die Variable  $i$  nicht inkrementiert wird, endet dieser Code niemals. Man bleibt in der Schleife hängen. Schlussendlich ist die Semantik dieses Codes nicht korrekt, weil dieser gar keinen Sinn macht.

Ce code ne finit jamais car la variable  $i$  n'est jamais incrémentée. On ne sort ainsi jamais de la boucle. On peut ainsi dire que c'est un programme sémantiquement incorrect, car il n'a pas vraiment de sens.

**Question | Frage 4**

Gegeben sei der folgende Code:

Soit la portion de code suivante :

```

1  int foo = 4;
2  for(foo = 10; foo >= 0; foo = foo - 3) {
3      System.out.println(foo);
4  }
```

- (a) *Welches Resultat wird von dieser Schleife angezeigt?*  
 Quel est le résultat affiché de cette boucle ?

**Solution:**

10  
7  
4  
1

- (b) *Wieviele Male wird diese Schleife ausgeführt?*  
 Combien de fois est exécutée la boucle ?

**Solution:** 4 Mal (und nicht 3, weil der Zähler *i* bei 10 startet).  
 4 fois (et non pas 3 car le compteur *foo* commence à 10 et se termine à 0).

- (c) *Schreiben Sie ein Program, welches das exakt gleiche Resultat liefert, unter Verwendung einer while Schleife.*  
 Écrire un programme strictement équivalent en utilisant une boucle *while*.

**Solution:**

```

1  int i = 10;
2  while(i >= 0) {
3      System.out.println(i);
4      i -= 3;
5  }
```

- (d) *Schreiben Sie ein Program, welches das exakt gleiche Resultat liefert, unter Verwendung einer do Schleife.*  
 Écrire un programme strictement équivalent en utilisant une boucle *do*.

**Solution:**

```

1  int i = 10;
2  do {
3      System.out.println(i);
4      i -= 3;
5  } while(i >= 0);
```

**Question | Frage 5**

Gegeben sei der folgende Code: | Soient les déclarations suivantes:

```

1  int a = 10;
2  int b = 20;
3  int c = ++a + b--;
```

Was sind die Werte von *a*, *b* et *c* nach der Ausführung diesem Code?  
 Quelles sont les valeurs de *a*, *b* et *c* après l'exécution de ce code ?

**Solution:** a = 11, b = 19 and c = 31

### Question | Frage 6

*Gegeben sei der folgende Code:*

Soit la portion de code suivante :

```
1  int a = 0; int b = 1; int c = 0;
2  for(int i = 0; i < 10; i++) {
3      a++;
4      b++;
5      c+=2;
6  }
```

*Geben Sie den Wert am Ende der Schleife von a, b und c?*

Quelles sont les valeurs à la fin de la boucle des variables a, b et c ?

**Solution:** *Der Wert von a ist 10, von b ist 11 und von c ist 20.*  
a vaut 10, b vaut 11 et c vaut 20.

### Question | Frage 7

*Schreiben Sie den Code ohne die Anweisung break:*

Écrire le code suivant sans instruction break :

```
1  int i = 0;
2  while(true) {
3      i++;
4      System.out.println(i);
5      if(i > 20) {
6          break;
7      }
8  }
```

**Solution:**

```
1  for(int i = 1; i <= 21; i++) {
2      System.out.println(i);
3  }
```



*Bemerkung: Die Sequenz \*- wird sechsmal wiederholt und mit jeder Iteration steigt die Anzahl der Sterne.  
Pour vous aider, remarquez que l'on répète la séquence \*- six fois et que le nombre d'étoiles augmente à chaque répétition.*

**Solution:**

```

1  for(int i = 1; i <= 6; i++) {
2      // Draw the stars
3      for(int j = 0; j < i; j++) {
4          System.out.print('*');
5      }
6
7      System.out.print('-');
8  }
9
10 // Carriage return at the end
11 System.out.println();

```

- (f) *Alle gerade Nummern im Interval  $\in [333 \dots 389]$  aufzuzählen.  
D'énumérer tous les nombres pairs dans l'intervalle  $\in [333 \dots 389]$ .*

**Solution:**

```

1  for(int j = 333; j <= 389; j++) {
2      // If number is even, display it
3      if((j % 2) == 0){
4          System.out.println(j);
5      }
6  }

```

*Andere Lösung mit einem while :*  
*Autre solution, avec un while :*

```

1  int current = 333;
2  int end = 389;
3
4  // Warning, here 389 should be included
5  while(current <= end) {
6      // If the current number is even, display it
7      if(current % 2 == 0) {
8          System.out.println(current);
9      }
10
11     // Increment the iterator
12     current++;
13 }

```