

Nom et prénom :

Classe :

TEST INTERMÉDIAIRE

SOLUTION

Informatique 1

Consigne :

Lisez attentivement la donnée et répondez de manière **lisible** aux questions. Vous avez droit pour cet examen à un aide-mémoire de 1 pages (1 feuille recto). Aucun moyen électronique n'est permis.

Un conseil : ne restez pas bloqués sur une question. Répondez tout d'abord aux questions avec lesquelles vous êtes à l'aise et revenez ensuite aux questions posant problème.

Le barème indiqué est indicatif.

Question	Points	Bonus	Score
Short questions	11	0	
Conditions	4	0	
Loops analysis	6	0	
Writing functions	10	0	
Working with strings	6	0	
Anagrammes	8	0	
Total:	45	0	

Cet examen a 6 questions, pour un total de 45 points.

Rev 1.02 Ω

Question 1 – Short questions (11 points)

Cette question est séparée en plusieurs exercices indépendants. Le nombre de point pour chaque exercice est indiqué dans la marge.

[3 Pt] (a) Soient les déclarations suivantes :

```
1 double a = 3.14;
2 short b = (short) 0xbeee;
3 char c = 'c';
4 char d = (char) 0b11;
5 String e = "foo";
```

Indiquez si les expressions suivantes sont correctes. Si elles sont correctes, donnez le type et la valeur (dans la base la plus logique) des expressions suivantes. Pour rappel, la valeur de 'a' est 97.

1) $a + c$

1) double, 102.14

2) $0xffff \& b + 0b1$

2) int, 0xbeef

3) $e + ((char)(c + (d \gg 1)))$

3) String, food"

4) $(c * a * e)$

4) Impossible

5) $a + (3 / a / b)$

5) Impossible

6) $c > d ? \text{"correct"} : \text{"wrong"}$

6) String, correct"

[2 Pt] (b) Lesquelles de ces expressions sont valides si x, y et z sont des entiers ?

1) $x \% y \% z == 0$

1) correct

2) $x < y < z$

2) incorrect

3) $y \% z == 0 \wedge x < y$

3) incorrect

4) $y * z / x < 0 \& z - y / x \geq 0$

4) incorrect

[1 Pt] (c) Citez deux différences entre String et char.

Solution: A char can only contain a single character whereas a String can contain several characters. A String is a class whereas a char is a primitive type. String literal values are typed with a double quote whereas a char literal value is entered with a single quote.

[4 Pt] (d) Vrai ou faux ? Réponse correcte +0.5, pas de réponse ou réponse fausse 0 points.

Les fonctions qui ne retournent rien ne doivent pas indiquer de type de retour.

True | False
 |

On peut appeler des fonctions qui sont déclarées en dessous de la ligne actuelle.

True | False
 |

L'opérateur ++ modifie la valeur de la variable sur laquelle il est appliqué.

True | False
 |

Deux fonctions différentes qui portent le même nom peuvent avoir le même nombre d'arguments.

True | False
 |

On peut écrire une boucle infinie à l'aide d'un for.

True | False
 |

La fonction main d'un programme peut retourner une valeur.

True | False
 |

`true && (true ^ true)`

True | False
 |

Les conversions explicites sont possibles uniquement lorsqu'il n'y a pas de perte de précision.

True | False
 |

[1 Pt] (e) Que valent j, k et i après l'exécution du code suivant ?

```
1 int i, j, k;
2 i = 9; j = 5; k = 25;
3 k /= --j - -i++;
```

Solution: i = 10, j = 4, k = 1

Question 2 – Conditions (4 points)

Écrivez le code correspondant aux conditions suivantes :

[1 Pt] (a) Les 8 derniers bits de l'entier x sont nuls.

Solution: `(x << 24) == 0 or (x & 0xff) == 0`

[1 Pt] (b) La variable de type double nommée $x \in [10 \dots 10^{24}]$.

Solution: `x > 10 && x < 1e24`

[1 Pt] (c) x est un multiple de $y + 1$ ou y est un multiple de $x + 2$.

Solution: `x % (y + 1) == 0 || y % (x + 2) == 0`

[1 Pt] (d) x et y , deux booléens, sont tous les deux vrais ou tous les deux faux.

Solution: `x == true && y == true || x == false && y == false`
 Other solution `x == y`

Question 3 – Loops analysis (6 points)

Que vont afficher les boucles suivantes sur la console ?

(a)

```
1  int j = 6;
2  for (int i = 0; i != j; j--) {
3      System.out.print(i + " " + j + " * ");
4      i++
5  }
```

Solution:

```
Solution:
0 6 * 1 5 * 2 4 *
```

(b)

```
1  int n, limit = 20;
2  for (n = 25-limit; n < limit; limit++) {
3      System.out.print(n + ", ");
4      System.out.println(limit);
5      n += 6;
6  }
```

Solution:

```
Solution:
5, 20
11, 21
17, 22
```

(c)

```
1  for (int i = 0; i < 3; i++) {
2      System.out.print(i + " ");
3      for (int j = 0; j < 3; j++) {
4          System.out.print(j + " ");
5      }
6      System.out.print(" * ");
7  }
```

Solution:

```
Solution:
0 0 1 2 * 1 0 1 2 * 2 0 1 2 *
```

Question 4 – Writing functions (10 points)

[3 Pt]

(a) Écrivez une fonction bar qui retourne true si deux nombres passés en argument sont positifs et qu'aucun des deux est pair. Dans les autres cas, la fonction doit retourner false.

Solution:

```
1  public static boolean bar(int a, int b) {
2      boolean result = a > 0 && b > 0;
3
4      if (result && a % 2 == 1 && b % 2 == 1)
5          return true;
6      else
7          return false;
8  }
```

- [3 Pt] (b) Écrivez une fonction qui convertit un nombre de jours (par simplification, nous considérons ici uniquement des jours complets) en secondes. Si le nombre de jours est négatif, votre fonction doit retourner un résultat positif.

Solution:

```

1 public static int convert(int nDays) {
2     int result = nDays > 0 ? nDays : -nDays;
3     result = result * 3600 * 24;
4     return result;
5 }
```

- [4 Pt] (c) Écrivez une fonction prenant une lettre en argument et qui retourne la lettre suivante de l'alphabet (ou la première lettre de l'alphabet pour z). Votre fonction ne devra accepter que les lettres minuscules. Pour toutes les autres lettres, elle devra retourner la valeur littérale '?'.

Solution:

```

1 public static char nextLetter(char x) {
2     if (!(x >= 'a' && x <= 'z'))
3         return '?';
4     else
5         if(x == 'z')
6             return 'a';
7         else
8             return (char)(x+1);
9 }
```

Question 5 – Working with strings (6 points)

Considérez que vous avez à disposition dans votre code les fonctions suivantes:

```

1 /** Returns the length (number of letters) of the String. */
2 public static int stringLength(String s);
3
4 /** Returns the char at position pos in s. */
5 public static char stringCharAt(String s, int pos);
```

- [2 Pt] (a) Écrivez une méthode `shortest` qui reçoit deux chaînes de caractères en argument et qui retourne la plus courte des deux. Si les deux chaînes ont la même longueur, la seconde est renvoyée.

Solution:

```
1 public static String shortest(String a, String b)
2 {
3     return stringLength(a) >= stringLength(b) ? b : a;
4
5     /* Other solution */
6     if(a.length() == b.length())
7         return b;
8     else
9         return a.length() < b.length() ? a : b;
10 }
```

- [4 Pt] (b) Écrivez une fonction qui retourne si un `String` passé en argument est composé de lettres dans l'ordre alphabétique. Par exemple, `abdos` est dans l'ordre alphabétique alors que `lapin` ne l'est pas. Pas mesure de simplicité, on considère que le `String` ne contient que des lettres minuscules. ⚠ Attention aux mots d'une seule lettre !

Solution:

```
1 public static boolean alpha(String x) {
2     if (x.length() <= 1)
3         return true;
4
5     for (int i = 0; i < x.length() - 1; i++) {
6         if (x.charAt(i) > x.charAt(i + 1))
7             return false;
8     }
9
10    return true;
11 }
```

Question 6 – ANAGRAMMES (8 points)

Dans cet exercice, on vous demande de mettre au point un détecteur d'anagrammes très utile pour le *Scrabble*TM. Deux mots sont des anagrammes si et seulement si toutes les lettres du premier mot se retrouvent dans le deuxième mot et réciproquement. Par exemple, les mots *lapin* et *alpin* sont des anagrammes.

- [6 Pt] (a) Pour réaliser cela, on vous demande d'écrire tout d'abord une fonction qui reçoit deux `String` et qui retourne si toutes les lettres du premier mot passé en argument se retrouvent dans le deuxième mot. Vous pouvez réutiliser les fonctions de `String` déjà présentées dans cet examen.

Solution:

```

1  public static boolean anagram(String a, String b) {
2      for(int i = 0; i < a.length(); i++) {
3          char currentLetter = a.charAt(i);
4          boolean found = false;
5
6          for(int j = 0; j < b.length(); j++) {
7              if(b.charAt(j) == currentLetter) {
8                  found = true;
9                  break;
10             }
11         }
12
13         if(!found)
14             return false;
15     }
16
17     return true;
18 }

```

- [2 Pt] (b) Complétez le code ci-dessous qui devra faire appel à votre fonction pour lire depuis la console deux mots¹. Vérifiez ensuite si ces deux mots sont des anagrammes et affichez un message indiquant dans chacun des deux cas si les mots sont des anagrammes ou non. ⚠ Certaines lettres peuvent se répéter !

```

1  public static void main(String[] args) {
2      String a =
3
4      String b =
5
6
7
8
9
10
11
12
13
14
15
16 }

```

Solution:

¹Cette partie est indépendante et vous pouvez y répondre si vous n'avez pas réussi à faire la fonction `anagram`.

```
1 public static void main(String[] args) {
2     String a = Input.readString();
3     String b = Input.readString();
4
5     // We have to check in both directions for cases such as "balle" & "belat"
6     // which would yield true for anagram("balle", "belat") but not for anagram("belat",
7     //     balle")
8     if(anagram(a, b) && anagram(b, a))
9         System.out.println("Those are anagrams");
10    else
11        System.out.println("Those are not anagrams");
12 }
```

Fin|Ende
