

Introduction à la Programmation (IP1)

Partiel – Durée : 2 heures

Université Paris-Diderot – Samedi 24 octobre 2015

- Aucun document ni aucune machine ne sont autorisés. Les téléphones doivent être rangés.
- Les exercices sont tous indépendants.
- Une réponse peut utiliser les réponses attendues à une question précédente (même si elle est non traitée).
- Les fragments de code Java doivent être correctement indentés.
- Dans les énoncés, on propose parfois une liste de fonctions et procédures utilisables. Vous pouvez cependant utiliser tout équivalent JAVA. (Par exemple, `System.out.print` à la place de `println`.)

Exercice 1

1. Écrire une fonction « `public static int nbMinutes (int d, int h, int m)` » qui prend en paramètres un nombre positif `d` de jours, un nombre positif `h` d'heures et un nombre positif `m` de minutes et qui renvoie le nombre de minutes qui se sont écoulées pendant ces `d` jours, `h` heures et `m` minutes. Par exemple, `nbMinutes (2, 10, 3)` s'évalue en 3483 puisque $2 * 24 * 60 + 10 * 60 + 3 = 3483$.
2. Écrire une procédure « `public static void showTime (int h, int m, int s)` » qui prend en paramètres un nombre positif `h` d'heures, un nombre positif `m` de minutes et un nombre positif `s` de secondes, et qui affiche sur une même ligne le nombre d'heures `h`, la chaîne « : », le nombre de minutes `m`, la chaîne « : » et le nombre de secondes `s`. Par exemple, l'appel de la procédure `showTime (12, 1, 37)` affiche 12:1:37.

Fonctions et procédures pouvant être utilisées :

- « `public static String intToString (int x)` » qui renvoie une chaîne représentant l'entier `x`.
- « `public static void printString (String msg)` » qui affiche la chaîne `msg`.

□

Exercice 2

1. Qu'écrire à la place de `A`, `B`, `C` pour que le fragment de code suivant compile correctement ?

```
1 public static A f (B a, C y) {
2     return (a[y] * 2);
3 }
```

2. Quelle est la valeur des variables `a`, `b` et `c` à la fin de l'exécution des instructions suivantes ?

```
1 int a = 0;
2 int b = 1;
3 int c = 2;
4 if (a == 2 || ! (b == 0)) {
5     c = 0;
6 } else {
7     c = 1;
8 }
```

□

Exercice 3

1. Qu'écrire à la place de `A`, `B`, `C` et `D` pour que la fonction `max3` renvoie la valeur maximale parmi les valeurs de `x`, `y` et `z` ? Par exemple, l'expression « `max3 (31, 13, 73)` » doit s'évaluer en 73. Ne justifiez pas votre réponse.

```
1 public static int max3 (int x, int y, int z) {
2     int m = 0;
3     if (x > y) {
4         if (x > z) {
5             m = x;
6         } else {
7             m = A;
8         }
9     } else {
```

```

10 |         if ( B ) {
11 |             C ;
12 |         } else {
13 |             m = z ;
14 |         }
15 |     }
16 |     return D ;
17 | }

```

2. En utilisant la fonction `max3` précédente, écrire une fonction « `public static int max5 (int x, int y, int z, int u, int v)` » qui calcule la valeur maximale parmi les valeurs de `x`, `y`, `z`, `u` et `v`. Par exemple, l'expression « `max5 (31, 13, 73, 37, -1)` » doit s'évaluer en 73. Ne justifiez pas votre réponse.

□

Exercice 4

- Écrire une fonction « `public static int nbZero (int [] t)` » qui prend en paramètre un tableau d'entiers et qui renvoie le nombre de cases de `t` valant 0. Par exemple, si `t` vaut `{1,0,2,0,0}`, `nbZero (t)` vaut 3.
Fonctions et procédures pouvant être utilisées :
— « `public static int intArrayLength (int [] t)` » qui renvoie la longueur du tableau `t`.
- Écrire une fonction « `public static int [] range (int start, int stop)` » qui prend en paramètres deux entiers positifs `start` et `stop` tels que `stop > start` et qui renvoie un tableau contenant tous les entiers de `start` (compris) à `stop` (compris). Vous n'avez pas à vérifier que `stop > start`, vous devez simplement le supposer. Par exemple, `range (3, 7)` vaut `{3, 4, 5, 6, 7}`.

□

Exercice 5

- Écrire une procédure « `public static void chessBoard (int w, int h)` » qui prend en paramètres deux entiers positifs `w` et `h` et qui affiche un échiquier de largeur `w` et de hauteur `h`. (Les cases blanches sont représentées par les O et les cases noires par les X.) La case en haut à gauche doit être un X. Par exemple, l'exécution de `chessBoard (5, 4)` affiche :

```

XOXOX
OXOXO
XOXOX
OXOXO

```

Fonctions et procédures pouvant être utilisées :
— « `public static void printString (String msg)` » qui affiche la chaîne `msg`.
— « `public static void newLine ()` » qui passe à la ligne suivante.
- Écrire une fonction « `public static boolean noRepetition (int [] t)` » qui prend en paramètre un tableau d'entiers `t` et qui renvoie `true` si et seulement si `t` ne contient pas deux fois la même valeur.
Par exemple, si `t` vaut `{ 73, -37, 33, 0, 33 }`, l'expression « `noRepetition (t)` » s'évalue en `false`.
Fonctions et procédures pouvant être utilisées :
— « `public static int intArrayLength (int [] t)` » qui renvoie la longueur du tableau `t`.

□

Exercice 6

- Écrire une fonction « `public static int nbChange (int [] t)` » qui prend en paramètre un tableau d'entiers et qui renvoie le nombre de fois que les valeurs des cases du tableau changent quand on le lit de la première à la dernière case. Par exemple, si `t` vaut `{ 1, 1, 1, 5, 3, 3, 3, 3, 1, 99, 99 }`, `nbChange (t)` vaut 4.
Fonctions et procédures pouvant être utilisées :
— « `public static int intArrayLength (int [] t)` » qui renvoie la longueur du tableau `t`.
- Écrire une fonction « `public static int [] describe (int [] t)` » qui prend en paramètre un tableau d'entiers `t` et qui renvoie un tableau d'entiers qui décrit le contenu de `t` si on le lit de la première à la dernière case. Cette description est la séquence des répétitions que l'on croise pendant cette lecture de gauche à droite du tableau. Pour décrire une répétition, on utilise deux entiers consécutifs `k` et `v` qui signifient que `v` apparaît `k` fois de suite dans le tableau `t`. Par exemple, si `t` vaut `{ 1, 1, 1, 5, 3, 3, 3, 3, 1, 99, 99 }`, `describe (t)` renvoie le tableau `{ 3, 1, 1, 5, 4, 3, 1, 1, 2, 99 }` puisque 1 apparaît d'abord trois fois de suite puis 5 apparaît une fois puis 3 apparaît quatre fois, 1 apparaît une fois et enfin 99 apparaît deux fois. (Pensez à utiliser la fonction `nbChange`.)
Fonctions et procédures pouvant être utilisées :
— « `public static int intArrayLength (int [] t)` » qui renvoie la longueur du tableau `t`.

□