Séance 4 de travaux pratiques

1 Exercices *

Sont-ce des facteurs (*?*) boucleavecdépendance conditionnelledansboucle tableaudeint liretableau paramètretableau [q1]

Compléter la fonction `allFactors` qui attend un tableau d'entiers et un entier et vérifie si tous les élements du tableaux sont des diviseurs de l'entier `n`.

La fonction renverra 1 si tous les éléments sont des diviseurs, et 0 sinon.

. `Factors.java`

Création d'un tableau (*?*) boucleavecdépendance tableaudeint définirtableau [q2]

Compléter la fonction `newArray` qui prend en paramètre un entier `n` et renvoie le tableau d'entiers de taille n dont la case d'indice 0 contient 0, la case d'indice 1 contient 1...

On rappelle qu'un tableau de taille n commence à l'indice 0, pour finir à l'indice `n-1`.

. `NewArray.java`

A□che le tableau ! (*?*) tableaudeint liretableau boucleavecdépendance paramètretableau [q3]

Compléter la procédure `showArray` qui attend un tableau d'entiers et qui a□che les éléments du tableau en colonne. Par exemple, si le tableau est `{42,3,1}`, la procédure a□chera :

```
42
3
1
```

. `ShowArray.java`

Sommes de factorielles (*?*) bouclesimbriquées boucleavecaccumulateur [q4]

Compléter la procédure `sumFactorials` qui attend un entier `n` et la somme des factorielles de 1 à `n`.

On rappelle que la factorielle d'un entier k est définie ainsi : $k! = 1 \times 2 \times \cdots \times k$. Il faut calculer $1 + 1 \times 2 + 1 \times 2 \times 3 + \cdots + 1 \times 2 \times \cdots \times n$.

. `SumFactorials.java`

Somme des entiers d'un tableau (*?*) boucleavecaccumulateur tableaudeint liretableau paramètretableau [q5]

Compléter la fonction `addUp` qui attend un tableau d'au moins 1 entier et renvoie la somme de ses éléments.

. `SumOfArray.java`

2 Exercices **

Retourner Max (*??*) boucleavecaccumulateur tableaudeint liretableau paramètretableau [q6]

Compléter la fonction `getMax` qui attend un tableau d'au moins 1 entier et renvoie le maximum.

. `MaxOfArray.java`

Travail sur tableau (*??*)

boucleavecaccumulateur tableaudeint liretableau définirtableau paramètretableau [q7]

Écrivez la fonction `minAvgMax`, prend un tableau en paramètre, calcule le minimum, la moyenne, et le maximum de ses valeurs, et renvoie un tableau contenant ces trois entiers (dans cet ordre-là).

. `MinAvgMax.java`

Prochains trains (*??*)

tableaudeint conditionnellecomplexe conditionnelledansboucle boucleavecdépendance liretableau [q8]

Compléter la procédure `showNextTrains` qui prend en paramètres `hour`, l'heure actuelle, `minute`, les minutes actuelles, `trainHours`, le tableau des heures des horaires des trains (dans l'ordre croissant), `trainMinutes`, le tableau des minutes de ces mêmes horaires et qui a□che les horaires des trains à l'heure actuelle ou après.

Par exemple, si

```
hour = 5;
minute = 13;
trainHours = {4, 5, 5, 6, 7};
trainMinutes = {7, 13, 17, 22, 4};
```

la procédure devra a□cher

```
05:13
05:17
06:22
07:04
```

. `NextTrains.java`

Cherche l'intrus ! (*??*)

boucleavecdépendance conditionnellecomplexe conditionnelledansboucle tableaudeint paramètretableau [q9]

Compléter la fonction `getOddOneOut` qui attend un tableau d'au moins 3 entiers qui sont tous égaux sauf 1, et qui renvoie l'intrus (l'entier di□érent des autres).

Par exemple, si le tableau est `{1, 1, 42, 1, 1, 1}`, la fonction renverra `42`.

. `OddOneOut.java`

Lire des tableaux dans tous les sens (*??*)

tableaudeint boucleavecdépendance liretableau paramètretableau [q10]

Complétez le code de la procédure `showArrays`, qui prend en paramètre trois tableaux d'entiers `t1`, `t2` et `t3`, afin qu'elle a□che le contenu de `t1`, puis celui de `t2` mais à l'envers (donc en commençant par son dernier élément, et en finissant par son premier), puis en lisant `t3` à l'endroit.

Par exemple, `afficheTableaux({1,2,2},{4,5,6},{9,8})` a□chera

```
1
2
2
6
5
4
9
8
```

. `ShowArrays.java`

Un carré vide (*??*) bouclesimbriquées conditionnelledansboucle boucleavecdépendance [q11]

Compléter la procédure `showSquare` qui attend un entier `size` et a□che un carré de coté `size`, avec des étoiles. Par exemple, pour `size = 4` :

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

. `StarSquare.java`

3 Exercices ***

Il lui manque une case (*???*) boucleavecdépendance bouclesimbriquées conditionnelledansboucle modulo dessin [q12]

« La classe, c'est d'être chic dans sa manière de s'habiller. Rien de tel que d'aller chez Azzedine Alaia ou même de s'acheter des sous-pulls chez Yohji Yamamoto. »

La mode de cette année est le damier, mais vous n'avez pas assez d'argent pour vous payer ces habits hors de prix. À défaut de classe, vous vous contenterez de remplir la méthode `drawCheck` de façon à ce qu'elle dessine un damier de `n` cases de côté, où `n` est le paramètre de `drawCheck`.

. `Checkboard.java`

PGCD d'un tableau (*???*) tableaudeint boucleavecaccumulateur bouclesimbriquées conditionnelledansboucle modulo liretableau paramètretableau [q13]

Compléter la fonction `getCommonDivisor` qui a pour paramètre un tableau d'au moins un entier et qui renvoie le PGCD de ces entiers (supposés positifs ou nuls) dans leur ensemble, autrement dit le plus grand entier positif qui divise tous les éléments du tableau.

. `CommonDivisor.java`

Un carré plein de carrés pleins de carrés... (*???*) bouclesimbriquées boucleavecdépendance conditionnelledansboucle conditionnellecomplexe modulo [q14]

Compléter la procédure `showSquare` qui attend un entier `size` et a□che un carré de coté `size`, avec des étoiles, plein d'autres carrés concentriques plus petits. Par exemple, pour `size = 8` :

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

. `FullStarSquare.java`

Nombres narcissiques (*???*) boucleavecdépendance boucleavecaccumulateur tableaudeint liretableau paramètretableau [q15]

Compléter la fonction `isNarcissistic` qui attend un tableau d'entiers qui contient dans l'ordre les chi□res d'un entier (e.g, 1,3,5 est le tableau qui représente l'entier 135) et qui renvoie 1 si cet entier est narcissique. (En mathématiques récréatives un entier n est narcissique si la somme de ses chi□res élevés à la puissance du nombre de ses chi□res donne l'entier n lui-meme. Par exemple 153 a 3 chi□res, et est narcissique car $1^3 + 5^3 + 3^3 = 153$).

. `NarcissisticNumber.java`

Un étrange carré ! (*???*) String boucleavecdépendance bouclesimbriquées [q16]

Compléter les procédures afin que :

-`getBase3(n)` renvoie la décomposition en base 3 de l'entier `n` à l'envers (c'est plus facile en partant de la fin), sous la forme d'une chaîne de caractères,

-`checkTwin(m,n,b)` renvoie `1` si le chi□re `b` apparait à la meme position dans les décompositions en base 3 des entiers `n` et `m`, `0` sinon,

-`showWeirdSquare(n)` a□che un carré de taille `n` qui contient à la ligne `i` et en colonne `j`, soit un espace si `i` et `j` ont un `1` à la même position dans leurs décompositions en base 3, soit un dièse.

Par exemple, `getBase3(12)` renverra la chaîne tableau `"011"` et `showWeirdSquare(3)` a□chera

```
###
# #
###
```

. `WeirdSquare.java`

4 Bac à sable

Test (*?*) [q17]

Faites vos tests ici !

. `Test.java`