Séance 3 de travaux pratiques

1 Exercices *

Une barre d'étoiles (⋆) String boucleavecaccumulateur [q1]

Compléter la fonction `getStarBar` qui attend un entier `n` et renvoie une chaîne de caractères contenant `n` fois le caractère `*`.

▷ `StarBar.java`

Puissances entières (⋆) boucleavecaccumulateur conditionnelledansboucle [q2]

Complétez la fonction `power` qui prend comme paramètres deux entiers positifs `x=base` et `y=exponent` et renvoie la valeur du premier monté à la puissance du deuxième. On suppose pour cet exercice que `y` est positif ou nul.

On rappelle que `x` à la puissance `y` vaut `x * x * x * ... * x`, `y` fois. Par convention, n'importe quel entier à la puissance 0 vaut 1.

▷ `PositivePower.java`

Les entiers pairs entre deux bornes (⋆) boucleavecdépendance conditionnelledansboucle modulo [q]

Compléter la procédure `showEvenInBounds` qui attend deux entiers `n` et `m` et affiche les entiers pairs entre `n` et `m` (séparés entre eux par `*`), suivi d'un retour à la ligne.

Si `n > m`, la procédure n'affichera rien.

Ainsi, pour `n=3` et `m=10`, la procédure affichera : `4*6*8*10*`

▷ `EvenInterval.java`

Calculer la somme de carrés d'entiers (⋆) boucleavecdépendance int [q4]

Compléter la fonction `computeSumOfSquares` qui attend un entier `n` (que l'on supposera supérieur ou égal à `1`) et renvoie le résultat de l'opération $1^2 + 2^2 + 3^2 + ... + n^2$.

▷ `SumOfSquares.java`

Tic-Tac (⋆) boucleavecdépendance conditionnelledansboucle [q5]

Compléter la procédure `showTicTac` qui attend un entier `n` et égrène `n` pas d'horloge en affichant `"tic"` sur les instants pairs, `"tac"` sur les instants impairs.

On pourra utiliser les fonctions `showTic` et `showTac`.

Pour `n=5`, votre code affichera :

```
tic
tac
tic
tac
tic
```

▷ `TicTac.java`

2 Exercices **

Les yeux dans les étoiles (⋆⋆) String boucleavecaccumulateur bouclesimbriquées [q6]

Compléter la fonction `getStarTriangle` qui attend un entier `n` et renvoie un triangle d'étoiles de hauteur et largeur `n`. Par exemple, `getStarTriangle(5)` renverra la chaîne de caractères suivante :

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

▷ `StarTriangle.java`

Puissances entières, le retour. (⋆⋆) boucleavecaccumulateur conditionnellecomplexe casparticuliers [q7]

Complétez la fonction `power` qui prend comme paramètres deux nombres et renvoie la valeur du premier monté à la puissance du deuxième. Attention à bien traiter le cas où le deuxième paramètre est négatif.

▷ `Power.java`

Conversion Binaire -> Décimale (⋆⋆) boucleavecaccumulateur String int [q]

Compléter la procédure `intOfBin` qui prendra en paramètre une chaîne de caractères contenant un nombre écrit en binaire, et renverra l'entier correspondant à l'écriture décimale de ce nombre.

Par exemple, `intOfBin("1010")` renverra `10`.

▷ `IntOfBin.java`

NSA H4xx0rZ (⋆⋆) boucleavecaccumulateur String [q9]

Nous allons implémenter le fameux algorithme de chiffrement ROT13. Cet algorithme, variante d'un code attribué à Jules César himself, est d'une simplicité redoutable. Il consiste à attribuer à chaque lettre de l'alphabet un numéro de 1 à 26, et à décaler une par une chaque lettre du message de 13 crans dans cette numérotation. Par exemple, le mot `hacker` devient `unpxre`, car `h = 8` et `u = 21 = 8 + 13`.

Votre mission, si vous l'acceptez, consiste à coder cet algorithme. Comme d'habitude, si votre code venait à être buggué, le gouvernement nierait avoir eu connaissance, etc.

▷ `ROT13.java`

3 Exercices ***

Un tapis d'étoiles (⋆⋆⋆) String boucleavecaccumulateur bouclesimbriquées [q10]

Compléter la fonction `getStarCarpet` qui attend deux entiers `m` et `n` et renvoie un rectangle de hauteur `m` et de largeur `n`, dont le bord est en étoiles `*` et l'intérieur en tiret `-`. On fera commencer cette chaîne par un retour à la ligne.

Pour exemple, `getStarCarpet(4,4)` renverra la chaîne de caractères contenant le rectangle suivant :

```
****
*--*
*--*
****
```

▷ `StarCarpet.java`

Fibonacci (⋆⋆⋆) int boucleavecaccumulateur [q11]

Compléter la procédure `showFibonacci` qui attend un entier `n` et qui affiche le `n`-ème terme de la suite de Fibonacci.

On rappelle que la suite de Fibonacci est définie ainsi par récurrence : $u_1 = 1$, $u_2 = 1$ et $u_{n+2} = u_{n+1} + u_n$ pour tout entier $n \geq 1$.

▷ `Fibonacci.java`

Paires diviseuses (⋆⋆⋆) boucleavecaccumulateur String int conditionnelledansboucle [q12]

Compléter la procédure `dobleDivide` qui prendra en paramètre un entier `n` et renvoie une chaîne de caractères contenant toutes les paires `(a,b)` telles que `a*b` divise `n`, séparées entre elles par un espace. Ces paires seront rangées dans l'ordre lexicographique, et l'on considera que les paires `(a,b)` et `(b,a)` sont identiques, on ne prendra donc que la première des deux (dans l'ordre lexicographique) en compte.

Pour exemple, `dobleDivide(8)` renverra la chaîne : `"(1,1) (1,2) (1,4) (1,8) (2,2) (2,4) "`.

▷ `DobleDivide.java`

Mieux vaux tartan que jamais (⋆⋆⋆) int conditionnellecomplexe boucleavecdépendance bouclesimbriquées conditionnelledansboucle dessin [q1]

Le fier clan des McAronee vous a confié la délicate tâche de produire leurs kilts. Pour ce faire il vous faudra programmer la Jacquard-2000™, un métier à tisser dernier cri afin qu'il vous produise la quantité de tartan nécessaire à leur commande. Petty Poy, le chef du clan, vous a fourni un échantillon de tartan. Votre but, écrire le code qui génère ce tartan.

Faites bien attention à le reproduire à la perfection, car il ne faut jamais froisser un Petty Poy écossais!

▷ `Kilt.java`

4 Bonus

Années bissextiles (⋆⋆) boucleavecdépendance conditionnelledansboucle conditionnellecomplexe modulo [q14]

Compléter la procédure `showBissextile` qui attend deux entiers `n` et `m` et affiche les années bissextiles entre l'année `n` et l'année `m` (bornes incluses). Chaque année est affichée sur une nouvelle ligne

Une année est dite bissextile si elle ou bien divisible par 4 mais pas par 100, ou bien divisible par 400.

Ainsi, pour `n=1992` et `m=2014`, la procédure affichera :

```
1992
1996
2000
2004
2008
2012
```

▷ `Bissextile.java`

5 Bac à sable

Test (⋆) [q15]

Faites vos tests ici!

▷ `Test.java`