Exercicis IIP Segon Parcial - Bucles (Tema 6)

P2 - Curs 18/19: *2 punts*

Un número poligonal és un número natural que pot recompondre's en un polígon regular de l costats. Per exemple, el número 9 és un número quadrat o el número 6 és un número triangular:

```
o o o       o
o o o      o o
o o o     o o o
```

En general, el n–èsim número poligonal es pot obtindre amb la fórmula:

$$\frac{n * [(l-2) * n - (l-4)]}{2}$$

on l és el número de costats del polígon. Per exemple, per a $l = 3$, els números 3-poligonals (triangulars) són 1 3 6 10 15 21 28...

Es demana: escriure un mètode estàtic que, donats un número k ($k > 0$) i el número de costats del polígon l ($l > 2$), torna `true` si k és un número l–poligonal i `false` en cas contrari. Per exemple, si $k = 15$ i $l = 3$, el mètode torna `true` (`15` és el 5-èsim número 3-poligonal) però, si $k = 19$ i $l = 3$, el mètode torna `false` (`19` no és un número 3-poligonal).

RecP2 - Curs 18/19: *2 punts*

Donat un enter $n \geq 0$, es desitja calcular l'invertit de n, és a dir, un altre enter que continga les mateixes xifres que n però en ordre invers. Per a això, s'escriuran un parell de mètodes que se suposaran en la mateixa classe, de manera que un puga usar a l'altre en els seus càlculs.

Es demana:

1. (1 punt) Realitzar un mètode estàtic que calcule el nombre de xifres d'un enter donat ≥ 0. Per exemple, per a 2347 el mètode ha de tornar 4, per a 8 ha de retornar 1, per a 0 ha de tornar 1.
2. (1 punt) Usant el mètode anterior, realitzar un mètode estàtic que calcule l'invertit d'un enter donat ≥ 0. Per exemple, per al 2347 haurà de calcular el 7432.

   Noteu en l'exemple que $7432 = 7 \cdot 1000 + 4 \cdot 100 + 3 \cdot 10 + 2 \cdot 1$.

   Es podrà usar el mètode `Math.pow(a, b)` predefinit de Java, que torna un `double`: $\mathtt{a}^{\mathtt{b}}$.

P2 - Curs 17/18: *2 punts*

Per a $n \geq 0$, les funcions enteres exponencial k^n i factorial $n!$, es poden calcular respectivament per les següents recurrències:

$$a_0 = 1 \qquad\qquad b_0 = 1$$
$$a_n = k \cdot a_{n-1}, \quad n > 0 \qquad\qquad b_n = b_{n-1} \cdot n, \quad n > 0$$

La funció factorial b_n creix més ràpidament que l'exponencial a_n, és a dir, a partir d'un cert n (major com més gran siga k), els termes $b_n > a_n$.

Es demana: escriure un mètode estàtic que reba com a paràmetre un enter $k > 1$, i que mostre en l'eixida, línia a línia, els successius termes:

a_0 b_0
a_1 b_1
a_2 b_2
...
fins a mostrar la primera línia en la qual el factorial supera a l'exponencial. Per exemple, per a $k = 3$, el mètode hauria d'escriure:

```
1       1
3       1
9       2
27      6
81      24
243     120
729     720
2187    5040
```

En la solució no es podran usar mètodes de la llibreria `Math` de Java.

RecP2 - Curs 17/18: *2 punts*

Es demana: implementar un mètode estàtic què, donat un nombre enter `n` $\geq$ `3`, escriga en l'eixida estàndard una figura de `n` línies. En cada línia s'han d'escriure tres caràcters `'N'` amb la separació adequada per tal que semble la lletra N majúscula. Per exemple, per a `n = 5` s'ha d'escriure:

```
NN    N
N N   N
N  N  N
N   N N
N    NN
```

P2 - Curs 15/16: *1.75 punts*

Es demana escriure un mètode Java estàtic que, donat un enter n $\geq$ 2, escriga en l'eixida una figura de n línies, amb dues diagonals que es junten en l'última línia, sobre un fons rectangular de `'-'`, de manera que:

- En cada línia s'han d'escriure dues `'V'` separades per un nombre de `'-'` cada vegada menor,
- en la primera línia la primera `'V'` apareix pegada al marge esquerre, i la segona en l'extrem dret de la figura.

Exemple per a n=5:

```
V--------V
-V------V-
--V----V--
---V--V---
----VV----
```

RecP2 - Curs 15/16: *1.5 punts*

Es demana: escriure un mètode de classe (`static`) que, donat un `String s`, mostre per pantalla un dibuix amb tantes línies com caràcters té l'`String`, amb dues diagonals que s'uneixen en la primera línia i divergeixen cap a l'última línia (això és, com una 'V' invertida), de manera que, en la línia i-èsima es mostre, en cada diagonal, el caràcter i-èsim de l'`String` donat. Pots suposar que la llargària de l'`String s` és major que 1.
NOTA: l'exemple es mostra dins d'una quadrícula per tal que pugues distingir millor els blancs del dibuix.

P.e., per a `s` = `"Java"`:

			J	J			
		a			a		
	v					v	
a							a

P2 - Curs 16/17: *2 punts*

Siga un enter `n` $\geq$ 2. **Es demana:** implementar un mètode estàtic que, per a tots els enters entre `2` i `n` inclusivament, torne un `String` amb la llista dels seus divisors propis. Recorda que els *divisors propis* d'un enter són tots els seus divisors excepte ell mateix i la unitat. Per exemple, per a `n = 18`, el mètode ha de tornar el següent `String`:

```
Divisors propis de 2:
Divisors propis de 3:
Divisors propis de 4: 2
Divisors propis de 5:
Divisors propis de 6: 2 3
Divisors propis de 7:
Divisors propis de 8: 2 4
Divisors propis de 9: 3
Divisors propis de 10: 2 5
Divisors propis de 11:
Divisors propis de 12: 2 3 4 6
Divisors propis de 13:
Divisors propis de 14: 2 7
Divisors propis de 15: 3 5
Divisors propis de 16: 2 4 8
Divisors propis de 17:
Divisors propis de 18: 2 3 6 9
```

RecP2 - Curs 16/17: *2 punts*

Siga un enter `a` $>$ `1`. **Es demana:** implementar un mètode estàtic que, usant `'*'`, mostre per pantalla una figura composada per un triangle rectangle isòsceles d'altura `a` i la seua imatge especular, encarats per la hipotenusa i amb les seues bases separades per un espai en blanc. Per exemple, per a `a = 4`, el mètode ha de produir la següent figura:

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

P2 - Curs 14/15: *1.75 punts*

Es diu que un número enter positiu és perfecte si és igual a la suma de tots els seus divisors (excepte ell mateix). **Es demana:** implementar un mètode de classe (o estàtic) que comprove si un enter `n`, `n` $>$ `0`, és un número perfecte. Per exemple, si `n` és `28`, el mètode ha de tornar `true` donat que els seus divisors són `1`, `2`, `4`, `7`, `14`, la suma dels quals val `28`.

RecP2 - Curs 14/15: *1.75 punts*

El postulat de Bertrand s'enuncia així: "Si n és un número natural major que 3, aleshores sempre existeix un número primer p tal que $n < p < 2 * n - 2$". En una certa classe es disposa d'un mètode amb perfil:

```
/** n > 1 */
public static boolean esPrimer(int n)
```

que, donat un `n` major que 1, comprova si és un número primer.

Es demana: implementa un mètode estàtic (que s'escriuria dins de la mateixa classe) que, donat un número natural `n` tal que `n` $>$ `3`, determine, usant el mètode `esPrimer` anterior, quin número primer `p` compleix el postulat de Bertrand per a `n`. D'haver-ne més d'un, ha de tornar el més menut. Per exemple, per a `n = 8`, els números primers dins de l'interval `[9,13]` són el `11` i el `13`, per tant, el mètode ha de tornar `11`.