

# Examen UF2

## Convocatòria extraordinària juny 2018

### Exercici 1. (3,5 punts)

Has d'implementar un mètode (`detectaLletra()`) que rebi com a paràmetres un text, un caràcter i un enter.

La funció ha de retornar un String que contingui el mot "Sí" si el caràcter està contingut en el text el nombre de vegades que indiqui l'enter. En cas contrari ha de retornar "No".

Per tal de comprovar la validesa de la funció caldrà que implementeu un programa principal o main que cridi la funció passant aquests tres paràmetres i imprimint el valor retornat.

Exemple:

Text: "Hola que hase"

caràcter: 'a'

enter: 2

la funció ha de retornar i el programa ha d'imprimir finalment: "Sí"

Exemple 2:

text: "Nullius in Verba"

caràcter: 'b'

enter: 9

la funció ha de retornar i el programa ha d'imprimir finalment: "No"

### Valoració

- Implementar correctament `detectaLletra()` (1)
- Implementar correctament `main()` (0,75)
- Implementar correctament `comprovaMatriu()` (1)
- Ús correcte de mètodes i pas de paràmetres (0,5)
- Identació, comentaris i ús correcte de variables i paràmetres (0,25)

## Exercici 2. (3 punts)

Implementa una funció que rebi com a paràmetre una Array d'una sola dimensió de números naturals però en format String (str), per exemple: {"23", "34", "6", "779", "123", "7", "6"} i que retorni la suma només dels nombres parells ( esbrinaParell() ). Aquest valor ha de ser imprès des de la funció principal.

És obligatori que cada nombre de l'Array sigui convertit a Enter mitjançant una altra funció ( stringAEnter() ) que haureu d'implementar i cridar des de la primera, abans de comprovar si és o no parell.

Exemple: Per la taula anterior, la funció hauria de retornar, 46.

### Valoració

- Implementar correctament la funció principal (0,5)
- Implementar correctament esbrinaParell() (1)
- Implementar correctament stringAEnter() (1)
- Identació, comentaris i ús correcte de variables i paràmetres (0,5)

## Exercici 3. (3,5 punts)

Tenim aquesta taula:

1000	876	44	65	17	232
343	67	55	76	15	344
567	57	66	87	14	454
457	34	99	98	13	565
23	789	88	43	12	676
678	26	77	32	11	787

i volem saber quins d'aquests nombres són, a la vegada:

- a - senars
- b - més grans de 10 i més petits de 200
- c - primer (només té a ell mateix i al nombre 1 com a divisors)

i imprimir-los per pantalla.

Caldrà implementar una funció per a cada comprovació i anar cridant-les mentre recorrem la matriu principal.

Aquí teniu les matrius implementades en Java i Python. Feu copiar i enganxar i continueu a partir d'aquí.

Java	Python
<pre> public class Exercici3 {      public static void main(String[] args) {         int[ ][ ] taula = { {1000,876,44,65,17,2,32},                            {343,67,55,76,15,344},                            {567,57,66,87,14,454},                            {457,34,99,98,13,565},                            {23,789,88,43,12,676},                            {678,26,77,32,11,787},                            };          // a partir d'aquí, el teu codi....      } // fi main } // fi classe </pre>	<pre> #programa principal  #matriu de treball...  matriu = [ [1000,876,44,65,17,2,32],             [343,67,55,76,15,344],             [567,57,66,87,14,454],             [457,34,99,98,13,565],             [23,789,88,43,12,676],             [678,26,77,32,11,787],  ];  # a partir d'aquí, el teu codi.... </pre>

#### Valoració

- Funció detecció senar(1)
- Funció que detecte rang correcte (1)
- Funció primer (1)
- Detecció de tres característiques a l'hora (0,5)