

Exercici 5: Algorismes numèrics.

Lliurament:

UN ÚNIC FITXER (exercici5.py) QUE CONTINGUI EL CONJUNT DE FUNCIONS QUE S'HAN IMPLEMENTAT.

Criteris d'avaluació (per ordre d'importància):

- 1.- El programa ha de donar un resultat correcte obligatòriament.
- 2.- Ús adequat del llenguatge (fer servir if/while correctament, fer servir la col·lecció adequada, etc.): 60% de la nota.
- 3.- Bon estil de programació (fer una interfase d'usuari adequada, comentaris, etc.): 40% de la nota.

La forma *empírica* més corrent per a saber quant triga un determinat algorisme a fer una determinada tasca és usar les funcions de Python que ens permeten consultar el rellotge de l'ordinador (que dona el temps en microsegons). Per exemple, si volguéssim saber el temps empleat per calcular la suma dels factorials de tots els enters fins a 200 ho podríem fer amb la funció següent:

```
def temps():
    import time
    import math
    t1 = time.clock()
    a = 0
    for i in range(1,200):
        a = a + math.factorial(i)
    t2 = time.clock()
    print "\n"
    print "El temps de proces ha estat %0.3f ms" % ((t2-t1)*1000)
```

- Escriu la versió recursiva del càlcul de l'enèsim element de la seqüència de Fibonacci, **fib1**. L'algorisme ha de demanar a l'usuari quin terme *n* vol calcular i ha d'imprimir el seu valor i quan ha trigat en calcular-lo.
- El Màxim Comú Divisor (MCD) de dos valors es pot calcular amb l'algorisme d'Euclides. Començant amb els valors *m* i *n*, apliquem repetidament la fórmula $n, m = m, n \% m$ fins que *m* és 0. Llavors, *n* és el MCD de *m* i *n*. Escriu una funció, **mcd**, que calcula el MCD de dos nombres enters entrats per l'usuari.
- El sedàs d'Eratòstenes és un algorisme antic per cercar tots els nombres primers fins a un determinat enter. Va ser creat per Eratòstenes (276-194 aC) un matemàtic de l'Antiga Grècia.
Referència: http://ca.wikipedia.org/wiki/Sedàs_d'Eratòstenes

Algorisme:

1. Escriu una llista A amb els números des del 2 fins a l'enter més gran N que vulgueu calcular.
2. El primer nombre de la llista és un nombre primer. Anoteu-lo en una llista de nombres primers, B.
3. Esborreu de la llista A el primer nombre i els seus múltiples.
4. Si el primer nombre de la llista A és més petit que l'arrel quadrada de N, torneu al punt 2.
5. Els nombres de la llista B i els que queden a la llista A són tots els nombres primers cercats.

Escriu una funció, **era1**, que demani a l'usuari un nombre *n* i llavors usi aquest algorisme per imprimir tots els nombres primers menors o iguals que *n*.

- Escriu una funció, **era2**, que imprimeixi el temps que es triga a calcular mitjançant el sedàs d'Eratòstenes els nombres primers menors que 10.000.000 i quants nombres primers hi ha.
- Escriu una funció **factorp** que comprovi si un determinat nombre n és primer mitjançant la tècnica de la factorització i que imprimeixi quan temps ha trigat en calcular-ho (podeu usar el mètode explicat a <http://www.purplemath.com/modules/factnumb.htm>).
- Escriu una funció, **fermatp**, que comprovi si un determinat nombre n és primer mitjançant la tècnica de Fermat amb valors $a=2, 3$, i 5 (veure apunts de teoria). La funció ha d'escriure al final del test el resultat (*primer* o *compost*) i el temps que ha trigat.