FUNDAMENTOS DE COMPUTACIÓN

Grado en ingeniería en tecnologías industriales

Universidad de Cantabria

Examen extraordinario(laboratorio)

7 de septiembre del 2018

Nombre y apellidos:

---

1. [25 %] Una empresa dispone de una base de datos de sus trabajadores. Se dispone de la siguiente información para este mes almacenada en un archivo de datos:

DatosB.txt

```
Nombre completo  Sueldo(Euros)  Pagado(SI/NO)  Antigüedad(años)
Garcia Ana,      1200.01,       SI,            5
Iglesias Jesus,  1800.01,       NO,            10
Nieto Juan,      1000.01,       SI,            3
Gutierrez Maria, 2000.01,       NO,            8
Gomez Pedro,     1500.01,       NO,            6
```

Escribe un programa que permita responder a las siguientes preguntas:

1. ¿Cuál es la antigüedad de cualquier empleado en la empresa? El programa pide por teclado un nombre de empleado y muestra por monitor su antigüedad. Si el empleado no está en la lista dada solicitarlo indefinidamente hasta que lo esté.
2. ¿A cuánto asciende la cantidad a pagar (haya sido pagada ya o no) a todos los trabajadores este mes? Usar una función para calcular esa cantidad.
3. ¿Cuántos trabajadores no han sido pagados aún este mes? Escribe sus nombres en un archivo denominado *NoPagados.txt*, junto con sus sueldos.

```
Dame un empleado
Alan Turing
Alan Turing no es un empleado
Dame un empleado
Nieto Juan
La antigüedad de Nieto Juan es 3 años
La cantidad a pagar este mes es 7500.05 euros
```

2. [25 %] Se tiene un número N de bombillas y un número N de interruptores situados en un panel, que cambian el estado de las bombillas de encendido a apagado o viceversa según el estado de la bombilla y siguiendo este esquema:

   - El primer interruptor cambia el estado de las bombillas 1,2,3,..,i,.. hasta la bombilla N.
   - El segundo interruptor cambia el estado las bombillas 2,4,6,...,2i,.. hasta la bombilla N.
   - El tercer interruptor cambia el estado de la bombilla 3,6,9, ...,3i,.. hasta la bombilla N.

     ...
   - El interruptor N sólo cambia el estado de la bombilla N.

   Por ejemplo: si N=4 y representado el estado de las bombillas por [0,0,1,0], siendo 0 el estado apagado y 1 encendido, pulsando los interruptores 1,2,3,4 se pasaría por los siguientes estados:

   $[1,1,0,1] \rightarrow [1,0,0,0] \rightarrow [1,0,1,0] \rightarrow [1,0,1,1]$. Quedando la bombillas 1,3 y 4 en estado encendido.

   Y si N=5, y el estado de las bombillas es [1,0,1,0,1], pulsando los interruptores 1,2,3,4 y 5 se pasaría por los siguientes estados: $[0,1,0,1,0] \rightarrow [0,0,0,0,0] \rightarrow [0,0,1,0,0] \rightarrow [0,0,1,1,0] \rightarrow [0,0,1,1,1]$. Quedando la bombillas 3 ,4 y 5 en estado encendido.

   1. Escribir una función que dado una lista con el estado asociado a cada bombilla, devuelva la lista del estado final de las bombillas si pulsamos ordenadamente todos interruptores desde 1 hasta la longitud de la lista.
   2. Escribir un programa que dada la lista anterior, muestre por pantalla una lista con las posiciones de las bombillas que quedan encendidas.

3. [25 %] Una permutación es simplemente un nombre para reordenar. Así, las permutaciones de la cadena 'abc' son 'abc', 'acb', 'bac', 'bca', 'cab', y 'cba'. Observar que la propia cadena es una permutación de ella misma (la permutación trivial). En este ejercicio se pide escribir una función `testpermutacion` que tenga como entrada dos cadenas `s1` y `s2` y devuelva el booleano `True` si la cadena `s2` es una permutación de la cadena `s1` o el booleano `False` en caso contrario. Como requisito todos los caracteres de la cadena `s1` deben ser distintos.

```
>>> testpermutacion("acb","bca")
True
>>> testpermutacion("abc","baa")
False
>>> testpermutacion("abc","ba2")
False
```

4. [25 %] Este ejercicio está dedicado a la construcción de una clase denominada POLINOMIO, para hacer algunos cálculos con polinomios sobre los números reales, que tendrán dos atributos; por un lado, la lista de sus coeficientes, formadas por $n+1$ números $[a_0, a_1, \ldots, a_{n-1}, a_n]$ y, por otro, la variable `x` de tipo `str`.

   Un polinomio de grado n en la variable x es:

$$a_0 + a_1 x + a_2 x^2 + \cdots + a_{n-1} x^{n-1} + a_n x^n \longleftrightarrow [a_0, a_1, a_2 \ldots, a_{n-1}, a_n]$$

   Por ejemplo, el polinomio $p(x) = 3 - x + 5x^2 - x^4$ de grado 4 está identificado por la lista $[3, -1, 5, 0, -1]$ y la variable x.

   Se pide:

   - Crear la clase POLINOMIO y adaptar el método `__str__(self)` para mostrar por pantalla un polinomio de forma más agradable y no como una lista de números. Por ejemplo, el objeto

```
>>> P=Polinomio([3,-1,5,0,1],'x')

>>> print(P)
3-x+5x**2+x**4
```

   - Implementar los métodos sumar, restar y multiplicar dos polinomios, adaptando estas operaciones con los métodos especiales:

```
__add__(self,otro)
 __sub__(self,otro)
__mul__(self,otro)
```

     Recordamos que dados dos polinomios $f(x)$ y $g(x)$ de grados n y m respectivamente:
     $f(x) = a_0 + a_1 x + a_2 x^2 + \cdots + a_{n-1} x^{n-1} + a_n x^n$ y $g(x) = b_0 + b_1 x + b_2 x^2 + \cdots + b_{m-1} x^{m-1} + b_m x^m$
     Supongamos que $n \geq m$ (similarmente si se tiene que $m \geq n$), entonces la suma y la resta es:

$$f(x) + g(x) = a_0 + b_0 + (a_1 + b_1)x + (a_2 + b_2)x^2 + \cdots + (a_m + b_m)x^m + a_{m+1}x^{m+1} + \cdots + a_n x^n$$

$$f(x) - g(x) = a_0 - b_0 + (a_1 - b_1)x + (a_2 - b_2)x^2 + \cdots + (a_m - b_m)x^m + a_{m+1}x^{m+1} + \cdots + a_n x^n$$

     Y el producto

$$f(x)g(x) = c_0 + c_1 x + c_2 x^2 + \cdots + c_{n+m-1} x^{n+m-1} + c_{n+m} x^{n+m} \quad \text{donde} \quad c_k = \sum_{i+j=k} a_i b_j$$

```
>>> P=Polinomio([3,-1,5,0,-1],'x')
>>> Q=Polinomio([0,-1,0,1,1],'x')
>>> R=Polinomio([1,-2],'x')
>>> print(P+Q)
3-2x+5x**2+x**3
>>> print(P-Q)
3+5x**2-x**3-2x**4
>>> print(Q*R)
-x+2x**2-2x**3+x**4-2x**5
```