FUNDAMENTOS DE COMPUTACIÓN

Grado en ingeniería en tecnologías industriales

Universidad de Cantabria

Control examen

16 de diciembre del 2016

Nombre y apellidos:

Aspectos generales de la prueba:

- Crear una carpeta en el escritorio denominada *control_16diciembre*. Guarda todos los ejercicios en esa carpeta, uno archivo por cada ejercicio: *ejercicio1.py*, *ejercicio2.py* , *ejercicio3.py* y el archivo *NoPagados.txt* generado en el Ejercicio 1.
- Al finalizar la prueba, subir los 3 archivos por separado al directorio de moodle, situado al final de la página y denominado SUBIR CONTROL 16/12/2016.

1. [30 %] 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.

```
$ python3 ejercicio1.py
Dame un empleado
Nieto Juan
La antigüedad de Nieto Juan es 3 años
La cantidad a pagar este mes es 7500.05 euros
```

2. [30 %] Construir una función en Python denominada `trocear` con dos argumentos una cadena `s` y con un carácter `a`. Y devuelva la `<<tupla>>` con todos los elementos encontrados al dividir la cadena `s` por el carácter `a`.

```
>>> trocear("hola, adios, ayer",",")
('hola', ' adios', ' ayer')
>>> trocear("hola, adios, ayer","a")
('hol', ', ', 'dios, ', 'yer')
>>> trocear("hola, adios, ayer","y")
('hola, adios, a', 'er')
```

3. [40 %] Supongamos que deseamos trabajar con polinomios, que vendrán definidos por sus coeficientes. Utilizaremos entonces el tipo(class) `list`, y más concretamente, *listas* formadas por $n+1$ números $[a_0, a_1, \ldots, a_{n-1}, a_n]$ para definir un polinomio de grado n, con $a_n \neq 0$.

$$a_0 + a_1x + a_2x^2 + \cdots + a_{n-1}x^{n-1} + a_nx^n \longleftrightarrow [a_0, a_1, a_2, \ldots, a_{n-1}, a_n]$$

Por ejemplo,

$p(x) = 3 - x + 5x^3$ de grado 3 está definido por la lista $[3, -1, 0, 5]$

y el polinomio

$q(x) = 1 + x - 2x^2$ de grado 2 está definido por $[1, 1, -2]$

Podemos sumar y multiplicar polinomios $p(x) + q(x), p(x) \times q(x)$:

$$p(x) + q(x) = q(x) + p(x) = 4 - 2x^2 + 5x^3 \longleftrightarrow [4, 0, -2, 5]$$

$$p(x) \times q(x) = q(x) \times p(x) = 3 + 2x - 7x^2 + 7x^3 + 5x^4 - 10x^5 \longleftrightarrow [3, 2, -7, 7, 5, -10]$$

En general, recordamos como sumar y multiplicar dos polinomios:
$f(x) = a_0 + a_1x + a_2x^2 + \cdots + a_{n-1}x^{n-1} + a_nx^n$ y $g(x) = b_0 + b_1x + b_2x^2 + \cdots + b_{m-1}x^{m-1} + b_mx^m$ de grados $n \geq m$ respectivamente:

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

$$f(x) \times g(x) = g(x) \times f(x) = c_0 + c_1x + c_2x^2 + \cdots + c_{n+m-1}x^{n+m-1} + c_{n+m}x^{n+m} \quad \text{donde} \quad c_k = \sum_{i=0}^{k} a_i b_{k-i}$$

Se pide escribir dos funciones en PYTHON `suma` y `producto` para calcular la suma y el producto de dos polinomios.

```
>>> suma([3,-1,0,5],[1,1,-2])
[4.0,0.0,-2.0,5.0]
>>> producto([3,-1,0,5],[1,1,-2])
[3.0,2.0,-7.0,7.0,5.0,-10.0]
>>> suma([3,-1,2],[1,1,-2])
[4.0]
>>> producto([0,1,3],[2,1])
[0.0,2.0,7.0,3.0]
```