

FUNDAMENTOS DE SISTEMAS DE INFORMACIÓN : Curso 2015.16

BOLETIN DE EJERCICIOS:

Print, Operadores aritméticos, lóxico, e sentencia IF,

WHILE/FOR, e funcións

strings e listas

diccionarios, e ficheiros

NOTAS:

Cando no exercicio x, se pida que faga un programa en Python, garde o seu contido nun ficheiro "x.py" (onde x é o número de exercicio).

ENTREGA:

Os alumnos deben entregar por email (antonio.master.fsi@gmail.com) os programas que resolvan os problemas plantexados nos exercicios 5, 16, 26, 27, 29, 30, 33, **40,42,50**

ASUNTO: [FSI][nome e apelidos do alumno]

Os ficheiros adxuntos chamaranse x.py (onde x é o número de exercicio)

PRINT

1. Cree un programa Python que (cun único print) imprima por pantalla a mensaxe

"primeiro programa en python"

2. Modifique o programa anterior para que imprima o mesmo, pero se usen 4 chamadas á función print (¿como evita que se meta un salto de liña ao final?)
3. Modifique o programa do exercicio 1 para que cada palabra sexa unha cadea independente que se imprime cun print con formato:

print "formato" % (secuencia de palabras a imprimir separadas por comas)

4. Cree un programa que, usando un único print imprima por pantalla:

"expreso o número 123.6 en formato decimal <123> e en formato real <123.60000>"

Úsense os modificadores %d e %f para que, aínda que o valor que lle pasemos ao print sexa sempre un 123.6, este se amose como <123> e <123.60000> respectivamente

5. Cree un programa que, usando un único print imprima por pantalla:

```
Estes son os operadores aritméticos en python
[tabulador] '+' [tabulador] para a suma
[tabulador] '-' [tabulador] para a resta
[tabulador] '*' [tabulador] para a multiplicación
[tabulador] '/' [tabulador] para a división
```

(onde poño [tabulador] debe aparecer unha tabulación, non imprimas [tabulador] por pantalla)

OPERADORES ARITMÉTICOS

6. Cree un programa que, partindo do contido de dúas variables cun valor enteiro (p.ex. a=25, b=33) imprima por pantalla o valor dunha terceira variable resultante de realizar a operación $c = a$ **<operacion> b**. Teña en conta os operadores +, -, *, /, %, **

```
$python p1.py
a=21, b=10
Line 1 - Valor de c= a+b : 31
Line 2 - Valor de c= a-b : 11
Line 2 - Valor de c= a*b : 210
Line 2 - Valor de c= a/b : 2
Line 2 - Valor de c= a%b : 1
Line 2 - Valor de c= a**b: 16679880978201
Line 2 - Valor de c= a//b: 2
```

7. Modifique o programa anterior para que o programa lle pida os valores das variables a e b ao usuario (por teclado)
8. Modifique o programa anterior para que os datos se reciban desde a liña de comandos (#import sys.argv)
9. ¿Que resultados se obterán ao evaluar as seguintes expresións en python? . Primeiramente "en papel", pense cal será o resultado das seguintes expresións. Faga un programa python que o chequee

```
2+3+1+2
2+3*1+2
(2+3)*1+2
(2+3)*(1+2)
5+---+6
3+2--1**3
4.0**(1/2)+1/2
4.0**(1.0/2)+1/2
3e8/100
4**.5
```

10. Traduzca as seguintes expresións a Python e avalíelas

a) $2 + (3 \cdot (6/2))$	c) $(4/2)^5$	e) $(-3)^2$
b) $\frac{4+6}{2+3}$	d) $(4/2)^{5+1}$	f) $-(3^2)$

Resultado debe ser a) 11; b) 2; c) 32; d) 64; e) 9; f) -9.)

11. Cree un programa que lle pida 2 números enteiros ao usuario e nos devolva o resultado de dividilos entre sí.:
- Facendo unha división real
 - Amosando o cociente e resto enteiros.

Formatee a saída do programa para que teña esta pinta

```
Introduzca un número a: 8
Introduzca un número b: 3
-----
Vostede quere dividir 8 entre 3
[tabulador] a entre b [real]= 2.6666666666666665
[tabulador] a entre b [int] = 2, e resto 2
```

12. Cree un programa que lle pida ao usuario o seu nome, o seu peso, e a súa altura en centímetros, e lle calcule o seu índice de masa corporal: $IMC = \text{peso} / (\text{altura}^2)$, onde peso está en Kg, e a altura en metros

```
Hola, imos calcular o seu IMC:
¿Como se chama vostede?: Juan
¿Canto pesa? (en Kg): 80
¿Canto mide? (en cm): 180
-----
Gracias "Juan", o seu IMC é: 24.691358
```

$$IMC = \frac{\text{peso}(kg)}{\text{altura}^2(m)}$$

- a. Modifique o programa anterior para (usando a función round) que o IMC se amose só con 2 díxitos decimais: ¿Funcionou? Se non funcionou probe a pór %5.2f para imprimir o imc

13. ..

OPERADORES LÓXICOS

14. Tendo en conta a precedencia de operadores e as táboas de verdade dos operadores lóxicos trate de evaluar "en papel" as seguintes expresións:

```
True and True
False and True
1==1 and 2==1
"test"=="test"
1==1 or 1!=2
(5>=3) and 1==1
True or 1==1
5<6<8
"test" == "testing"
"test" != "testing"
"test" ==1
not (True and False)
not (1==1 and 0!=1)
not (10==1 or 1000==1000)
not (1!=10 or 3==4)
not ("testing" == "testing" and "Zed" == "Cool")
1==1 and not ("testing"= =1 or 1==0)
"comida" == "bebida" and not (3==4 or 3==3)
3==3 and not ("test" == "test" or
"programa"=="completo")
4<5<9
8>6<5
```

Cree un programa en Python para validar que as súas respostas son as correctas

15. Cree un programa que amose a táboa da verdade da operación XOR. Isto é, asumindo dúas variables enteiras x e y (que tomarán os valores 0 e 1). Amosaráse unha táboa como a seguinte

```
Esta é a táboa de verdade do XOR
  x    y    XOR
True  True  False
False True   True
True  False True
False False False
```

Non escriba directamente "True" nin "False" como unha cadea de caracteres!

Truco: se $x=1; y=0 \rightarrow x \wedge y = 1 \rightarrow \text{bool}(x \wedge y)=\text{True}$

SENTENCIA IF:

16. Cree un programa que lle pida ao usuario o seu nome, o seu peso, e a súa altura en centímetros, e lle calcule o seu índice de masa corporal: $IMC = \text{peso} / (\text{altura}^2)$, onde peso está en Kg, e altura en metros [Xa o fixemos antes]

Adicionalmente indique o estado do usuario en función do IMC (tal e como se reflexa na seguinte táboa)

IMC	ESTADO
ata 18.49	Baixo Peso
[18.5 a 24.99]	Normal
[25.5 a 29.99]	Sobrepeso
de 30 en adiante	Obeso

https://es.wikipedia.org/wiki/Indice_de_masa_corporal

```
Hola, imos calcular o seu IMC, e o seu estado:
¿Como se chama vostede?: Juan
¿Canto pesa? (en Kg): 80
¿Canto mide? (en cm): 180
-----
Gracias "Juan", o seu IMC é: 24.691358
O seu estado actual é: "Normal"
```

17. Cree un programa que reciba unha cantidade en euros (cantidade enteira) e desglose cantos billetes de 500, 200, 100, 50, 20, 10, 5. e moedas de 1 ou 2 euros son necesarios para

```
Introduzca unha cantidade en euros (sen decimais): 434
434 euros precisan de:
    2 billetes de 200 euros
    1 billetes de 20 euros
    1 billetes de 10 euros
    2 moedas de 2 euros
```

18. Cree un programa que reciba a nota numérica dun alumno en prácticas (entre 0 e 10) e no exame (entre 0 e 10) . E calcule a nota final asumindo que as prácticas valen o 20% da nota e o exame o 80% restante. Ambas notas deben ser introducidas desde a liña de comandos (primeiro a nota de prácticas e despois a nota do exame)

Asegúrese de que as notas introducidas están no rango [0..10]. Doutro xeito indique a mensaxe "a nota de prácticas non é válida" ou "a nota do exame non é válida" respectivamente. E nese caso non se amosa unha nota final por pantalla.

A saída do programa deber ter a seguinte "aparición"

```
$ python p3.py 5 8
A súa nota final é: 7.4000
```

19. Modifique o programa anterior para que se amose a calificación final do alumno en función da súa nota final.

```
$ python p3.py 5 8
A súa nota final é: 7.4000 (NOTABLE)
```

20. Cree un programa que lle pida ao usuario desde a liña de comandos 3 argumentos:

operando1 operando2 operador.

O programa realizará a operación : operando1 operador operando2

e amosará o resultado por pantalla (saída parecida á amosada abaixo)

- Os operadores posibles son +, -, * e /

- Asíumase que os números introducidos sempre serán enteiros

- O programa debe chequear que se a operación é unha división (/) o divisor sexa diferente de CERO

```
$ python p3.py 5 8 -
O resultado de operar 5 - 8 é: -3
$ python p3.py 4 8 +
O resultado de operar 4 + 8 é: 12
$ python p3.py 5 0 /
O resultado de operar 5 / 0 é: indeterminado
Non se pode dividir por 0!
$ python p3.py 4 8 /
O resultado de operar 4 / 8 é: 0
```

SENTENCIAS WHILE E FOR:

21. Cree un programa que pida un número por teclado, e calcule a súa factorial (utilice un while)

```
$ python p21.factorial.py
Introduzca un numero (para calcular factorial):50
factorial de 50 e:
30414093201713378043612608166064768844377641568960512000000000000
```

¿Pode usar un bucle FOR? Inténtelo!

22. Cree un programa amose os múltiplos de 3 entre o 1 e o 100 (incluidos salvo que se indique o contrario)

```
$ >python p22.multiplos3.py
amosando múltiplos de 3 entre [1 e 100]: 3 6 9 12 15 18 21
24 27 30 33 36 39 42 45 48 51 54 57 60 63 66 69
72 75 78 81 84 87 90 93 96 99
```

23. Cree un programa amose os múltiplos de 3 entre o 1 e o 50, e os múltiplos de 5 entre 51 e 100.
¿Pode usar un único bucle while?

```
$ python p23.multiplos3.5.py
amosando múltiplos de 3 entre [1 e 5]: e de 5 entre [51 e 100] 3
6 9 12 15 18 21 24 27 30 33 36 39 42 45 48 55 60
65 70 75 80 85 90 95 100
```

24. Cree un programa que amose os 10 primeiros números pares positivos

```
$ python p24.pares.py
amoso os 10 primeiros pares positivos: 2 4 6 8 10 12 14 16 18 20
```

25. Cree un programa que amose canto suman os números no rango [0,n]. Amose ese valor para todos os n entre [1 e 10]

```
$ python p25.suma10.py
A suma entre 0 e 1 e: 1
A suma entre 0 e 2 e: 3
A suma entre 0 e 3 e: 6
A suma entre 0 e 4 e: 10
A suma entre 0 e 5 e: 15
A suma entre 0 e 6 e: 21
A suma entre 0 e 7 e: 28
A suma entre 0 e 8 e: 36
A suma entre 0 e 9 e: 45
A suma entre 0 e 10 e: 55
```


26. Cree un programa que pregunte ao usuario un número enteiro, e nos diga se é "primo" ou é "non primo" (nese caso amose un dos seus divisores).

```
$ python p26.primo.py
Introduzca un numero entre 1 e 1000000: 15
o numero 15 non e primo, pois 3 e un divisor

$ python p26.primo.py
Introduzca un numero entre 1 e 1000000: 23
o numero 23 e primo
```

** forma trivial: ver se (desde o 2 ata o numero anterior hai algun numero que o divide)

27. **MENÚS:** Cree un programa que lle pida ao usuario por teclado que elixa unha entre as 5 opcións operación (+, -, *, / ou s) que se lle amosarán nun **menú interactivo**.

Se a opción é 's', o programa remata. Se a opción é '+', '-', '*' ou '/' pídelle 2 operandos ao usuario operando1 e operando2. A continuación fai a operación **operando1 opcion operando2** e amosa o resultado por pantalla (saída parecida á amosada abaixo)

- O programa debe chequear que se a operación é unha división (/) o divisor sexa diferente de CERO, en caso contrario, avisa da situación e non fai a división

```
$ python p27.menu.operadores.py

===== Calculadora en Python =====
Elixa unha operacion:
+ : para sumar
- : para restar
* : para multiplicar
/ : para dividir
s : sae do programa!
=====
opcion: +
Introduzca operando1: 5
Introduzca operando2: 2
O resultado de operar 5 + 2 é: 7

===== Calculadora en Python =====
Elixa unha operacion:
+ : para sumar
- : para restar
* : para multiplicar
/ : para dividir
s : sae do programa!
=====
opcion: s
Good bye!!
```

Pista: Temos un bucle que remata só cando o usuario introduce a opción 's'. E cada vez que se fai unha iteración fai os seguintes pasos: (1) amosa o menú de opcións, (2) le a opción elixida, (3) mira se debe saír (nese caso remata), ou debe proceder a ler operandos (4), nese executa a operación prevista coa opción e operandos introducidos [como no exercicio 20].

FUNCIÓNS

28. Modifique o programa 21, de xeito que haxa 1 función "factorial(n)" que devolve a factorial do número n. Úsese dita función

```
$ python p28.fact.py
Introduce un num para calcular factorial:5
factorial de 5 e 120
```

29. Modifique o programa 26, de xeito que haxa 1 función "e_primo(n)" que indique se o número é primo ou non, e ademais indique cal é o divisor de n en caso de que o haxa. A función devolverá sempre 2 valores: un booleano (é ou non primo), e un divisor que será 1 se o número é primo, ou d, tal que $n\%d=0$ (d é o primeiro divisor de n atopado)

```
$ python p29.primo.py
Introduzca un numero entre 1 e 1000000: 15
o numero 15 non e primo, pois 3 e un divisor

$ python p29.primo.py
Introduzca un numero entre 1 e 1000000: 23
o numero 23 e primo
```

30. Modifique o programa 27, de xeito que o programa sexa similar, pero agora haxa 2 funcións:
- funcion menu: sen arguments, amosa o menu e devolve a opcion que o usuario elixe
 - funcion executa: recibe como argumentos a operacion a realizar, e dous operandos: fai os chequeos pertinentes (operación '/') e imprime por pantalla o resultado

```
$ python p30.menu.operadores.py

===== Calculadora en Python =====
Elixa unha operacion:
+ : para sumar
- : para restar
* : para multiplicar
/ : para dividir
s : sae do programa!
=====
opcion: +
Introduzca operandol: 5
Introduzca operando2: 2
O resultado de operar 5 + 2 é: 7

===== Calculadora en Python =====
Elixa unha operacion:
+ : para sumar
- : para restar
* : para multiplicar
/ : para dividir
s : sae do programa!
=====
opcion: s
Good bye!!
```

31. Cree unha función que reciba dous números como parámetros e intercambie os seus valores. ¿É posible?

```
$ python p31.py
6 9
9 6
```

Truco, a función debe recibir 2 valores (a,b) e devolver 2 valores (b,a) e asignámosllos ás variables a,b de novo.

Nota, recorde que tamén pode facer directamente en python: b,a = a,b (asignación múltiple)

32. Cree unha función xeradora que se chame **primos** que devolva números primos entre 1 e un valor máximo que se indicará como parámetro (con valor por defecto = 1000000)
Utilice dita función para crear un programa que reciba desde a liña de comandos un número N, e amose todos números primos entre 1 e N (incluidos)

```
$python p32.xeradora.py 18
1
2
3
5
7
11
13
17
```

Este debe ser o programa principal:

```
##principal
from sys import argv
if len(argv) !=2:
    print "sintaxe incorrecta: use python %s <maxprimo>" % argv[0]
    exit()

max = int(argv[1])

for i in primos(max):
    print i
```

33. O seguinte código imprime por pantalla 3 liñas cos valores 4, 3, e 2

```
def subrutina(): #l.1
    def subrutina2(): #l.2
        global a #l.3
        print(a) #l.4
        a=2 #l.5
        return #l.6
    #l.7
    a = 3 #l.8
    subrutina2() #l.9
    print(a) #l.10
    return #l.11
    #l.12
### PRINCIPAL #l.13
a = 4 #l.14
subrutina() #l.15
print(a) #l.16
```

Explique brevemente por que o último print non escribe un 4 en pantalla

STRINGS...

35. Probe que fan as seguintes sentencias en Python

```
print "%.s." % 'hola'
print "%.8s." % 'hola'
print "%-8s." % 'hola'
print "%.d." % 16
print "%+d." % -16
print "%+d." % 16
print "% d." % 16
print "%.6.3f." % 16
print "%Xf." % 16
print "%#Xf." % 16
```

Probe tamén a escribir un '%%', un '%e'

36. Escriba un programa que escriba por pantalla a cadea amosada no seguinte cadro. Na medida do posible use os operadores + e ** para evitar escribir a mesma subcadea varias veces (é dicir, non faga simplemente un print " asdfasdfasdf ==- -== -==- -==- -==- -==- -==- asdfasdf ")

```
asdfasdfasdf ==- -== -==- -==- -==- -==- -==- asdfasdf
```

37. Cree unha función que chequee se unha cadea recibida desde a liña de comandos é un palíndromo

```
$ python p.35.palindromo.py ANA E ANA
"ANA E ANA e un palindromo"
$ python p.35.palindromo.py HOLA MUNDO
"HOLA MUNDO NON e un palindromo"
```

Notas, quizais:

- necesite usar " ".join(argv[1:]) para obter unha cadea a partir de argv
- crear unha función invirte(s), que dado un string s nos devolve unha copia da mesma pero escrita ao revés
 - impleméntea usando un bucle while e o operador + (para ir engadindo caracteres á cadea a devolver)
 - probe a implementala usando o operador de indexación s[::-1]

38. Corrixa a seguinte función "encuentra_char" para que busque a posición dun carácter dentro dunha cadea. Devolverá a posición dentro da cadea, ou -1 se non a atopa (similar a find). Comprobe o seu funcionamento nun programa python

```
# PROGRAMA p.38.ex.encuentraChar.py
# -*- coding: cp850 -*-
#busca carac dentro de cadena e devolve a posición onde
#o atopa ou -1 se non o atopa
def encuentra_char(cadena, carac):
    indice = 0
    while indice < len(cadena):
        if cadena[indice] == carac:
            return indice

    return 0

##main
texto = "alabar a la alabarda"
print texto
print encuentra_char(texto,"b")

$python p.38.ex.encuentraChar.py
alabar a la alabarda
3
```

39. Modifique o programa anterior para admita un parámetro opcional de xeito que se devolva a posición onde apareza un carácter, pero comezando a búsqueda a partires dunha posición inicial (por defecto posicion=0)

```
## cree a súa función encuetra_char_pos aquí
....

##programa principal
texto = "alabar a la alabarda"
print texto
print encuentra_char_pos(texto,"b")
print encuentra_char_pos(texto,"b",4)

$python p.39.ex.encuentraCharPos.py
alabar a la alabarda
3
15
```

40. Escriba o programa p.40.ex.quitarvocais.py, e nel cree unha función elimina_vocais que recibirá como argumento un string, e devolverá unha nova cadea onde todas as vocais foron eliminadas.

```
$python p.40.ex.quitarvocais.py
alabar a la alabarda
lbr l lbrd
```

Nota, pode ser de axuda crear unha función adicional es_vocal(caracter) ás que se lle pasará un string que contén unha única letra e nos indica se é unha vocal (True) ou se non é unha vocal (False)

41. Modifique o programa anterior de forma que en lugar de eliminar as vocais, haxa unha función `reemplaza_vocais(cadea, letra)`, de xeito que todas as vocais da cadea pasada como argumento sexan reemplazadas pola letra pasada como argumento.

```
$python p.41.ex.reemplazavocais.py
alabar a la alabarda
_l_b_r_l_l_b_rd_
$python p.41.ex.reemplazavocais.py
alabar a la alabarda
*l*b*r * l* *l*b*rd*
```

42. Escriba o programa `p.42.binario.py`, que pida ao usuario un número en binario (cadea con 1s e 0s) e nos diga o seu valor en base 10 (formato decimal)

```
$python p.42.binario.py
110010
valor decimal= 50
```

Nota, para pasar **100011** a decimal hai que coller o último dígito e multiplícalo por 2^0 , o resultado sumarllo ao (penúltimo multiplicado por 2^1), o resultado sumarllo ao (antepenúltimo multiplicado por 2^2), o resultado sumarllo ao (anterior multiplicado por 2^3), etc

É dicir $110010 = 0 \times 2^0 + 1 \times 2^1 + 0 \times 2^2 + 0 \times 2^3 + 1 \times 2^4 + 1 \times 2^5 = 50$

43. [DIFICULTADE ALTA] un programa que conte cantas palabras hai nun texto que lle pedirá ao usuario. Fágao de xeito que o programa principal chame a unha función `contar_palabras(texto)` que devolva o número de palabras que se atopan no texto. Asíumase que unha palabra é calquera secuencia de caracteres (alfanuméricos ou non) diferentes do espazo en branco

```
$ python p.43.contarpalabras.py
Introduzca un texto (enter para fin) s fsf sfs sfjs    === aaa
Hai 6 palabras
```

Comprobe se o a súa función funciona ben cando recibe tabuladores ou saltos de liña. Para iso engada as 3 liñas seguintes ao seu programa e comprobe o resultado (debe contar 5 palabras)

```
texto = "alabar \t a la alabarda\n \t \n alabada "
```

```
print texto
print "Hai ", contar_palabras(texto) , " palabras"
```

44. Probe os programas de exemplo que se deixaron nas transparencias relativas a "String: Métodos e funcións", e entenda que fan as funcións: `capitalize`, `lower`, `upper`, `title`, `isalnum`, `len`, `strip`, `count`, `find`, e `replace`
45. .
46. .
47. .
48. .
49. .

LISTAS

50. Cree un programa que lle pida a usuario una secuencia de números (ata que o usuario introduzca un 0). Cando o usuario remate, o programa amosará todos os números introducidos (excepto o 0 final) e a súa media.

```
$ python p.50.numeros.py
Introduzca un número (0 para fin) : 1
Introduzca un número (0 para fin) : 3
Introduzca un número (0 para fin) : 5
Introduzca un número (0 para fin) : 0
Vostede introduciu os números 1 3 5
A media deses números e: 3
```

Truco, comece cunha lista baleira, e vaia engadindo os números que lle introduzcan

51. Cree un programa que lle pida a usuario una secuencia de strings (ata que o usuario introduzca simplemente un [Enter]). Cando o usuario remate, o programa todos os strings ordeados alfabéticamente

```
$ python p.51.strings.py
Introduzca un nome (enter para fin) : Pepe Gotera [enter]
Introduzca un nome (enter para fin) : Pepe Villuela [enter]
Introduzca un nome (enter para fin) : Harrison Ford [enter]
Introduzca un nome (enter para fin) : [enter]
Vostede introduciu estas 3 cadeas:
Harrison Ford
Pepe Gotera
Pepe Villuela
```

52. Cree un programa (SIMILAR AO 43) que conte cantas palabras hai nun texto que lle pedirá ao usuario. Fágao de xeito que o programa principal chame a unha función **contar_palabras(texto)** que devolva o número de palabras que se atopan no texto

```
$ python p.52.contarpalabras.py
Introduzca un texto (enter para fin) s fsf sfs sfjs    === aaa
Hai 6 palabras
```

Teña en conta que "hola\n\n mundo\t".split() crea unha lista cos elementos resultantes de trocear a cadea orixinal, e onde os caracteres de corte son o espacio, o \t e o \n.

53. Cree un que usando un menú como o amosado abaixo, permita introducir froitas nunha lista de froitas, e tamén eliminar unha froita determinada, e amosar todas as froitas que se atopan na lista

```
$ python p53.nomes.py

===== Listado de froitas =====
Elixa unha opción:
+ : para engadir froitas
- : para eliminar
m : para amosar contido
s : sae do programa!
=====
opcion: +
Introduzca unha froita: uva
'uva' foi engadido á lista de froitas

===== Listado de froitas =====
Elixa unha opción:
+ : para engadir froitas
- : para eliminar
m : para amosar contido
s : sae do programa!
=====
opcion: +
Introduzca unha froita: laranxa
'laranxa' foi engadido á lista de froitas

===== Listado de froitas =====
Elixa unha opción:
+ : para engadir froitas
- : para eliminar
m : para amosar contido
s : sae do programa!
=====
opcion: m
A lista de froitas contén:
    uva
    laranxa

===== Listado de froitas =====
Elixa unha opción:
+ : para engadir froitas
- : para eliminar
m : para amosar contido
s : sae do programa!
=====
opcion: -
Introduzca unha froita: laranxa
'laranxa' foi eliminado da lista de froitas

===== Listado de froitas =====
Elixa unha opción:
+ : para engadir froitas
- : para eliminar
m : para amosar contido
s : sae do programa!
=====
opcion: s
Good bye!!
```

TUPLAS

60. Probe os exercicios propostos nas transparencias de clase. Salvo engadir elementos/modificalos (as tuplas son inmutables!!)... todo é moi similar ao uso de listas

DICCIONARIOS

70. Cree un programa que, usando diccionarios, permita crear un traductor de palabras de palabras de español a inglés, e teña as seguintes opcións

```
$python p70.axenda.py

===== DICCIONARIO Esp-Ing =====
Elixa unha opción:
+ : para engadir unha palabra ao diccionario
- : para eliminar unha palabra do diccionario
e : para amosar todas as palabras españolas
i : para amosar todas as palabras inglesas
m : para amosar todo o contido do diccionario
    (unha palabra e a súa tradución por liña)
t : traduce unha palabra do español ao inglés
s : sae do programa!
=====
```

- programa debe comezar creando un diccionario baleiro $D=\{\}$ ao que se lle irán engadindo pares que conteñan unha palabra en español e outra en inglés. Por exemplo nun momento dado o diccionario pode conter: $\{\text{'casa': 'house', 'pan', 'bread', 'ojo': 'eye'}\}$
- Créese unha función para amosar o menú (que faga os prints correspondentes, e devolva a opción elexida polo usuario).
- Créese unha función para cada unha das funcións do menú (excepto para 's'), de xeito que cando o usuario introduzca a opción:
 - + : se invoque a función engade(D)
 - : se invoque a función elimina(D)
 - e : se invoque a función esp(D)
 - i : se invoque a función ing(D)
 - m : se invoque a función amosarTodo(D)
 - t : se invoque a función traducir(D)

engade(D): nesta función pídenselle ao usuario unha palabra en español E, e outra en inglés I, e engádense ao diccionario D ($D[E] = I$)

elimina(D): nesta función pídeselle ao usuario unha palabra en español x, e elimínase do diccionario D tanto a palabra x como a súa tradución (del $D[x]$)

esp(D): usará $D.keys()$ para obter todas as palabras en español do diccionario e amosaráas por pantalla (unha por liña)

Ing(D): usará $D.values()$ para obter todas as palabras en inglés do diccionario e amosaráas por pantalla (unha por liña)

amosarTodo(D): usará $D.items()$ para obter todos os pares (pal_espagnol, pal_ingles) e amosaráas por pantalla (un par por liña)

traducir(D): pídelles unha palabra en español (E) ao usuario e amosa a súa tradución. ($get(E, None)$)

FICHEIROS

71. Engádalle ao programa anterior dúas opcións máis:

S: Grabar o contido do dicionario a un ficheiro chamado dict.dat

R: Restaurar o contido do dicionario desde o ficheiro dict.dat (o cal restaura as entradas do dicionario co contido previamente salvado en disco). É dicir, se no ficheiro había 5 palabras coas súas 5 traducións, tras "restaurar" o contido do dicionario, o noso programa debe conter un dicionario con 5 palabras.