

## Exercici 1 ( 2,5 punts).

Es crearà una funció que es dirà exercici1 i que rebrà únicament com a paràmetres:

- Un string del tipus “opcio1;opcio2;opcio3; .....; opcioN;” que haurà de tenir com a mínim 2 opcions.
- Una tupla que anomenarem exceptions i que tindrà com a valor per defecte una tupla amb l'element “A”.

La funció llançarà excepcions del tipus:

**ValueError** si el string d'opcions no és una string, amb el missatge associat "The options have to be a string"

**ValueError** si la tupla d'excepcions no és una tupla, amb el missatge associat "The exceptions have to be a tuple"

**ValueError** si el string d'opcions no té com a mínim 2 elements, amb el missatge associat "The minimum amount of options are 2".

I en qualsevol daquests 3 casos, la funció finalitzarà.

La funció mostrarà un menú amb les opcions del string, per exemple, si cridem a la funció com segueix:

```
options = "opcio1;opcio2;opcio3;"  
funcionA(options,exceptions=("c","d",10))  
la funció ens mostrarà el següent menú:
```

```
1) opcio1  
2) opcio2  
3) opcio3  
option:
```

És a dir, les opcions numerades començant per 1.

Si escollim una opció que no és un enter o no està en excepcions, llançarà una excepció del tipus **IndexError** amb el missatge associat, "The option is not a number and is not in exceptions", mostrarà una pausa “Enter to continue” i tornarà a mostrar el menú a l'usuari.

Si escollim una opció que és un número però està fora del rang (1 – 3) en aquest cas, i no està en excepcions, llançarà una excepció del tipus **IndexError** amb el missatge associat, "The option is out of range and not in exceptions”, mostrarà una pausa “Enter to continue” i tornarà a mostrar el menú a l'usuari.

En cas d'escollir una opció vàlida, ens tornarà:

- Si és una opció vàlida del menú, un sencer amb el número d'opció escollida.
- En cas d'escollir una opció vàlida de les excepcions, ens tornarem l'opció escollida en el mateix format que està en excepcions.

És a dir, si es crida a la funció com segueix:

```
options = "opcio1;opcio2;opcio3;"  
funcionA(options,exceptions=("c","d", 10, "11"))
```

en cas d'escollir un número de l'1 al 3, tornareu un sencer.  
En cas d'escollir c, d o 11, tornarà un string amb el valor escollit.  
En cas de triar-ne 10, de tornarà un sencer amb el valor 10.

## Exercici 2 ( 1 punt )

Es crearà una funció amb el nom `exercici2` que rebrà com a paràmetres entre 2 i n string, i ens tornarà un string amb tots els string passats, concatenats i separats per un “-”.

Aquesta funció llançarà una excepció del tipus **`TypeError`** en cas que algun dels paràmetres que passem no sigui un string, mostrarà un missatge descriptiu de l'error i finalitzarà la seva execució.

És a dir si cridem a la funció com segueix:

```
print(exercici2())  
print(exercici2("A"))
```

Donarà un error i acabarà la seva execució

si cridem a la funció com segueix:

```
print(exercici2("a","b",1))
```

Llançarà una excepció del tipus **`TypeError`**, mostrarà un missatge descriptiu i n'acabarà l'execució.

si cridem a la funció com segueix:

```
print(exercici2("a","b"))  
ens retornarà el string "a-b".
```

```
print(exercici2("a","b","c"))  
ens retornarà el string "a-b-c".
```

```
print(exercici2("a","b","c","d"))  
ens retornarà el string "a-b-c-d".
```

## Exercici 3 (2,5 punts)

Es crearà una funció amb el nom `exercici3`. Aquesta funció rebrà com a mínim 1 paràmetre que es dirà assignatura. Aquesta funció no retornarà res en cap cas.

Si cridem a la funció com segueix:

```
exercici3("Maths")
```

ens imprimirà:

```
=====Maths=====
```

Code	Midterm Exam	Final Exam	Average Grade
------	--------------	------------	---------------

i no retornarà res.

També podrà rebre tantes parelles com vulguem del tipus:

id=[float1,float2]

on

**id** representarà l'identificador de cada alumne, que serà del tipus XY345. És a dir, dues lletres majúscules i 3 números.

**float1** representarà la nota del parcial de l'assignatura i **float2** representarà la nota de l'examen final.

Les parelles del tipus id=[float1,float2] representaran les notes de l'alumne amb l'identificador id.

Si cridem a la funció com segueix:

exercici3("Maths",RT345=[2.05,5.04])

Ens mostrarà la següent taula:

=====Maths=====			
Code	Midterm Exam	Final Exam	Average Grade
RT345	2.0	5.0	5.0
=====			
Total Average			5.0

És a dir, les notes amb un sol decimal, la mitjana de les dues notes, amb un sol decimal, i al final, el càlcul de la mitjana de totes les notes finals, amb un sol decimal també.

Per arrodonir les notes a un sol decimal podeu fer servir la funció round.

round("número a arrodonir, "decimals que volem")

Si cridem a la funció com segueix:

exercici3("Maths",RT345=[7.15,6.55],SW432=[5.44,7.63])

Ens mostrarà la següent taula:

=====Maths=====			
Code	Midterm Exam	Final Exam	Average Grade
RT345	7.2	6.5	23.4
SW432	5.4	7.6	20.8
=====			
Total Average			22.1

Si cridem a la funció com segueix:

exercici3("Maths",RT345=[6.32,8.27],SW432=[5.44,9.77],OT441=[4.35, 6.18])

Ens mostrarà la següent taula:

=====Maths=====						
Code	Midterm	Exam	Final	Exam	Average	Grade
RT345		6.3		8.3		26.1
SW432		5.4		9.8		26.6
OT441		4.3		6.2		13.4
=====						
Total Average						22.0

Aquesta funció llençarà excepcions del tipus:

**ValueError** en el cas que els valors passats com id=valor, valor no sigui una llista mostrarà el missatge associat

"The values have to be a list of integers"

I finalitzarà la seva execució.

**ValueError** en el cas que els valors passats com id=valor, id no tingui el format correcte ( dues lletres i 3 números) mostrarà el missatge associat

"The id have to have the format: two Letters + 3 numbers"

I finalitzarà la seva execució.

**AssertionError** en el cas que els valors continguts a la llista que es passa com a valor no tingui longitud 2, mostrarà el missatge associat

"The list have to contain only two floats" i finalitzarà la seva execució.

**TypeError** en el cas que un dels dos valors continguts a la llista que es passa com a valor no siguin floats, mostrarà el missatge associat

"Values in list have to be floats" i finalitzarà la seva execució.

#### Exercici 4 ( 1,5 punts)

Es crearà una funció el nom de la qual serà exercici4, que rebrà com a únics paràmetres dues llistes de longituds aleatòries entre 0 i 10 cadascuna, de números aleatoris entre 0 i 15. I que ens tornarà una llista amb els números comuns a les dues llistes, sense repetició.(intersecció de llistes).

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

Es crearà una funció el nom de la qual serà exercici4, que rebrà com a únics paràmetres dues llistes de longituds aleatòries entre 0 i 10 cadascuna, de números aleatoris entre 0 i 15. I que ens tornarà una llista amb tots els números que hi ha en alguna de les llistes, i sense repetició. ( unió de llistes).

#### Exercici 6 ( 1,5 punts).

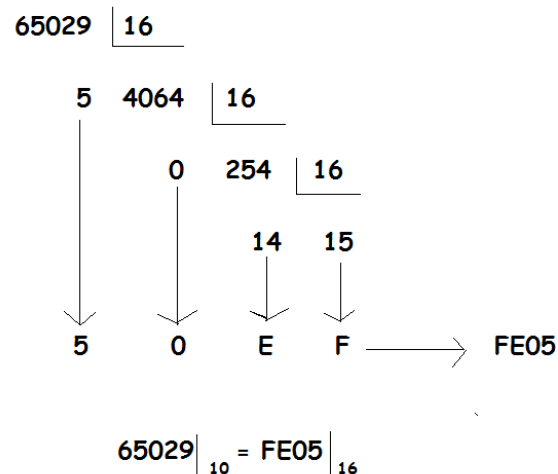
S'ha de crear una funció **recursiva** que converteixi un nombre decimal en su valor en hexadecimal (base 16).

Aquesta funció rebrà com a únic paràmetre el número que volem convertir i retornarà un **string** amb la representació en base 16 d'aquest número.

Nota: El canvi de base es realitza mitjançant divisions successives per 16 en les quals les restes determinen els dígit hexadecimals del número segons la següent correspondència:

Residu	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Dígit	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F

Per exemple:



Exemples de números en base 16:

48160 en base 16 = BC20  
 5132 en base 16 = 140C  
 46697 en base 16 = B669  
 38374 en base 16 = 95E6  
 29691 en base 16 = 73FB  
 26067 en base 16 = 65D3  
 20053 en base 16 = 4E55  
 34379 en base 16 = 864B  
 47551 en base 16 = B9BF  
 28492 en base 16 = 6F4C

Una posible solució iterativa seria:

```
def iterative_hexadecimal(num):
    hexadecimal_digits = [0, 1, 2, 3, 4, 5, 6, 7, 8, 9, "A", "B", "C", "D", "E", "F"]
    result = ""
    while num >= 16:
        result = str(hexadecimal_digits[num%16]) + result
        num = num // 16
    result = str(hexadecimal_digits[num]) + result
    return result
```

