



Exercici 1:

L'any 2001 A.C., el Cèsar Modulatix ha decidit realitzar un cens entre tots els habitants de l'imperi romà. Els ciutadans de Roma pertanyen a una de les següents **categories: poble, soldat, centurió o Cèsar**. Es desitja emmagatzemar la següent informació de cada persona: nom, regió, edat, sexe (H / D / I) i categoria. Es demana realitzar un programa en python per a la gestió d'aquest cens, per a partint de les següents dades:

```
censImperi={
"AR1":{"nom":"Claudia","region":"Roma","edat":23,"sexe":"I","categoria":"centurio"},
"AD3":{"nom":"Maximo","region":"Hispania","edat":30,"sexe":"H","categoria":"soldat"},
"AC2":{"nom":"Marco","region":"Hispania","edat":28,"sexe":"H","categoria":"soldat"},
"AV6":{"nom":"Julius","region":"Roma","edat":35,"sexe":"H","categoria":"cesar"},
"SS5":{"nom":"Augustus","region":"Hispania","edat":21,"sexe":"H","categoria":"soldat"},
"WQ6":{"nom":"Eugenia","region":"Hispania","edat":28,"sexe":"D","categoria":"centurio"},
"JU8":{"nom":"Anastacia","region":"Hispania","edat":17,"sexe":"I","categoria":"soldat"},
"DF5":{"nom":"Aurelia","region":"Hispania","edat":20,"sexe":"D","categoria":"poble"},
"BR1":{"nom":"Antistia","region":"Roma","edat":16,"sexe":"D","categoria":"centurio"},
"KR9":{"nom":"Apolonia","region":"Roma","edat":25,"sexe":"I","categoria":"centurio"},
"KH7":{"nom":"Acucia","region":"Roma","edat":29,"sexe":"D","categoria":"centurio"},
"XH4":{"nom":"Titus","region":"Roma","edat":39,"sexe":"I","categoria":"poble"},
"KA2":{"nom":"Aurelio","region":"Roma","edat":15,"sexe":"H","categoria":"poble"},
"MJ8":{"nom":"Tiberius","region":"Roma","edat":22,"sexe":"H","categoria":"poble"},
}
```

La clau del diccionari consta de dos lletres i un número i representa un codi únic que identificava a cada habitant de l'Imperi Romà.

```
categoria=["soldat","cesar","centurio","poble"]
```

```
regio=["Roma","Hispania"]
```

Descripció de les opcions a implementar:

1. MENU: (0.5 punts)

Es crearà una funció amb el nom menús (tupla, capsalera,títol), que rebrà dos paràmetres, un de les tuples amb el contingut d'un dels menús i una capçalera. Els paràmetres *capsalera* i *títol* seran opcional tenint com a valor per defecte una cadena buida.

La funció ens mostrarà el menú associat a aquesta tupla, per exemple, quan rebí la tupla:

```
datos=("Incloure persona en el cens","Buscar persona en el cens","Mostrar soldados Hispania","Visualitzar categoria del cens")
```

```
titol="CENS ROMA"
```

```
capsalera="_"
```



ens mostrarà el menú corresponent:

CENS ROMA
1 . Incloure persona en el cens
2 . Buscar persona en el cens
3 . Mostrar soldats Hispania
4 . Visualitzar categoria del cens
Escoje una opción:

I en retornarà l'opció escollida, comprovant que introduïm l'opció correcta.  
Aquesta funció llançarà excepcions en el cas que:

- S'introdueixi una opció no numèrica.
- S'introdueixi una opció que no apareix al menú.

I ens mostrarà un missatge descriptiu de l'error corresponent, tornant a mostrar el menú i demanant una opció per consola.

**2. INCLOURE PERSONA EN EL CENS: (1.5 punts)** Es demanarà a l'usuari el nom, el sexe, la regió, l'edat, la força i la categoria. Totes les dades hauran de ser vàlides, reintentant la seva lectura fins a que siguin vàlides.

Per controlar totes aquestes validacions, haurem de crear les següents funcions:

- Funció **enterFrase**: La funció rep un paràmetre per defecte que es "nom". Aquesta funció s'utilitzarà per comprovar que el nom, la categoria i la regió només estan formades per caràcters. Quan el paràmetre per defecte tingui el valor de nom, mostrarà un missatge demanant a l'usuari el seu nom. Quan el paràmetre per defecte sigui la regió mostrarà un missatge demanant a l'usuari la regió. La funció haurà de comprovar que la categoria i la regió només tenen els valors possibles de la llista categoria i regió. La funció printarà per pantalla un missatge d'error adequat al context, per exemple: "Aquesta regió no és vàlida",
- Funció **enterSexe**: Aquesta funció no rep paràmetres, demana a l'usuari el sexe i comprova que només pot ser H (home), D (dona) o I (intersexual). Quan sigui correcte, el retorna.
- Funció **enterCategoria**: Demana a l'usuari una categoria i comprova que només està format per caràcters. Els únics valors possibles són: **poble, soldat, centurió o Cèsar**. A més a més, només pot haver un **Cèsar** a l'imperi Romà.
- Funció **enterEdat**: Aquesta funció no rep paràmetres i retorna l'edat. Comprova que l'edat es un valor numèric entre 0 i 100.
- Funció **enterForça**: Aquesta funció no rep paràmetres i retorna la força de la persona. Assigna un valor aleatori decimal entre 50 i 200.
- Funció **idPersona**: Aquesta funció no rep paràmetres i retorna l'id de la persona. Li demana a l'usuari el id de la persona a censar i comprova que el id està format per dos lletres i un número. En cas contrari, informarà del error i tornarà a demanar un altre id. No poden haver id's repetits

**NOTA:** Totes les validacions de les funcions anteriors s'ha de comprovar amb l'estructura **try....except....**

Abans d'inserir a una persona s'haurà de tenir en compte que: una persona no pot estar censada dues vegades i no pot existir més d'un Cèsar en l'Imperi. Si es compleixen totes aquestes condicions s'inseriran les dades de la persona en el cens. Si s'incomplís alguna de les condicions anteriors es donarà un missatge d'error indicant aquesta situació, tornant el programa a menú principal. Exemples:

Nombre: Máximo  
Regió: Hipania  
Edat: 30  
Sexe: H  
Categoria: Soldat

Nom: Claudia  
Regió: Roma  
Edat: 120 **Edad Erronia (0-100)**  
Edat: 23  
Sexe: X **Sexe Erroni (H/M/I)**  
Sexe: I  
Categoria: centuriona **Categoria Errònia**  
Categoria: centurió



3. **BUSCAR PERSONA EN EL CENSO: (0.75 punts)** Aquesta funció s'anomenarà *buscarPersona*, Es demanarà el nom d'una persona, mostrant-se en pantalla totes les dades d'aquesta persona. Si la persona no ha estat censada, es donarà un missatge d'error indicant que no s'ha trobat la persona sol·licitada tornant el programa a menú principal. Exemples:

Introdueixi Nom: Marco Aurelio

Error: Marco Aurelio no ha estat censat.

Introdueixi Nom: An

DADES DEMANADES

NOM	REGION	EDAT	SEXE	CATEGORIA
Anastacia	Hispania	17	I	Soldat
Antistia	Roma	16	D	Centurio

Per realitzar el print de d'alt hem de fer una funció que s'anomeni *imprimir*, que ens imprimeixi per pantalla la informació amb el format de la imatge de d'alt.

```
imprimir(cabecera, datos, titulo="")
```

La funció *buscarPersona* farà una crida a la funció *imprimir* per tal de printar la informació amb el format adequat.

4. **MOSTRAR ELS SOLDATS D'HISPANIA: (0.5 punts)** La funció s'anomenarà *mostrarSoldatsHispania*. Funció que mostrarà els soldats de la regió d'Hispania ordenats en ordre decreixent d'edat. Aquesta funció s'implementarà utilitzant sort/sorted i/o funcions lambda. Per printar la informació s'utilitzarà la funció *imprimir* de l'apartat anterior.

SOLDADOS DE HISPANIA				
NOM	REGION	EDAT	SEXE	CATEGORIA
Maximo	Hispania	30	H	soldat
Marco	Hispania	28	H	soldat
Augustus	Hispania	21	H	soldat
Anastacia	Hispania	17	I	soldat

5. **VISUALITZAR CATEGORIA DEL CENS: (0.75 punts)** La funció s'anomena *mostrarCategoria*. Es demana a l'usuari una categoria (que haurà de ser correcta, reintentant la seva lectura fins que ho sigui), això implica cridar a la funció *enterCategoria*. Si la categoria es correcta, es mostrarà per pantalla tota la informació de les persones censades que pertanyen a aquesta categoria, fent una pausa després de cadascuna d'elles. Si no hi ha cap persona d'aquesta categoria es mostrarà per pantalla un missatge indicant-ho. A més a més, es mostrarà un missatge amb el recompte total de persones que hi ha a la categoria. Exemple:

Introdueixi la categoria: CENTURIO

Nom: Claudia

Regió: Roma

Edat: 23

Sexe: M

Categoria: centurio

Premi una tecla per a Continuar .....

No hi ha més persones censades d'aquesta categoria.

**Total de persones en la categoria CENTURIO : 5**



Exercici 2: (2 punts)

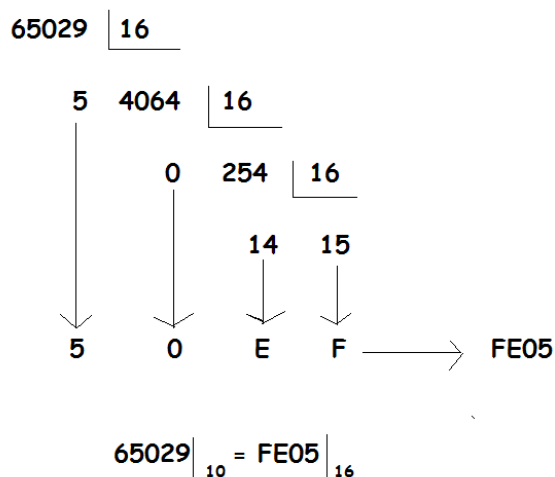
Construeixi una funció que converteixi un nombre decimal en el seu valor en hexadecimal (base 16).

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.

Recordatori: El canvi de base es realitza mitjançant divisions successives per 16 en les quals les restes determinen els dígit hexadecimal 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