

# Fonaments de Programació (104337)

Curs 2020-2021 - Examen Parcial (10 de Novembre de 2020)

Nom estudiant:

NIU:

**Important:** Recordeu que cal donar les millors solucions possibles en cada exercici. A més de funcionar correctament, els procediments i funcions han d'estar ben programats (utilitzant les instruccions més adients, sense operacions ni variables innecessàries, etc.)

**Els exercicis 1 a 8 fan referència a un mateix tema que introduïm a continuació:**

En un club esportiu ens demanen fer un programa per gestionar els seus socis. Per cada soci, es vol guardar el seu DNI i el seu any de naixement. Per tant, necessitarem dues tenir dues llistes per a guardar aquesta informació. Les dues llistes contenen la informació d'un mateix soci en les mateixes posicions. Per exemple, el DNI de la posició 3 de la llista de DNIs i l'any de la posició 3 de la llista d'anys de naixement corresponen al mateix soci.

Els socis del club paguen diferent en funció de la seva edat. Hi ha definides 4 categories de socis segons en quina franja d'edat es troben: 1-infantil, 2-juvenil, 3-adult i 4-sènior.

Per a fer aquest programa, cal fer una sèrie de funcions (exercicis 1 a 7) que implementen algunes de les tasques necessàries per a fer la gestió dels socis. Aquestes funcions s'utilitzaran per a fer el programa principal (exercici 8).

## Exercici 1 (1 punt)

Fer una funció anomenada `es_nombre` que rebi com a paràmetre una cadena de caràcters, i retorni `True` si tots els elements de la cadena són dígit (0..9) i `False` en cas contrari.

```
def es_nombre(cadena):
    i=0
    trobat=False
    while i<len(cadena) and not trobat:
        if cadena[i] <'0' or cadena[i] > '9':
            trobat=True
        else:
            i+=1
    return not trobat
```

## Exercici 2 (1 punt)

Fer una funció anomenada `dni_correcte` que rebi com a paràmetre una cadena de caràcters, i comprovi si té el format de dni correcte. Si el dni és correcte, la funció retornarà `True`, i en cas contrari `False`.

Un DNI és correcte si té 8 dígit (0 al 9) seguits d'un caràcter de control (lletra en majúscula, de la A a la Z). Per saber quina lletra correspon al dni, prenem el nombre complet de 8 xifres, el dividim entre 23 i ens quedem amb la resta (residu) de la divisió. La lletra s'assigna segons la taula següent:

Resta	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22
Lletra	T	R	W	A	G	M	Y	F	P	D	X	B	N	J	Z	S	Q	V	H	L	C	K	E

**Nota:** Per fer part d'aquesta funció, utilitzeu la funció `es_nombre` de l'exercici 1.

```
def dni_correcte(cadena):
    lletres=('T','R','W','A','G','M','Y','F','P','D','X','B','N','J','Z',
            'S','Q','V','H','L','C','K','E')
    if len(cadena) == 9:
        if es_nombre(cadena[:8]):
            residu=int(cadena[:8])%23;
            if cadena[8]==lletres[residu]:
                return True
            else:
                return False
        else:
            return False
    else:
        return False
```

### Exercici 3 (1 punt)

Fer una funció anomenada `afegir_element` que insereixi un element al final d'una llista. La funció rebrà una llista i un element a inserir. Abans d'inserir l'element, s'haurà de comprovar que NO hi sigui a la llista, i només s'inserirà si no hi és. La funció retornarà `True` si s'ha pogut inserir i `False` en cas contrari (ja hi era).

**IMPORTANT:** NO UTILITZAR LA INSTRUCCIÓ `in`

```
def afegir_element(llista,element):
    i=0
    trobat=False
    while i<len(llista) and not trobat:
        if llista[i]==element:
            trobat=True
        else:
            i+=1
    if trobat:
        sortida = False
    else:
        llista.append(element)
        sortida = True
    return sortida
```

#### Exercici 4 (1 punt)

Fer una funció anomenada `calcular_quota` que permeti saber la quota de soci a cobrar depenent de l'edat del soci. La funció rebrà dos paràmetres: la quota base i l'edat, i retornarà el valor de la quota a cobrar. En el cas que l'edat sigui menor a 16 anys, s'aplicarà una reducció del 20% de la quota base. En cas de que l'edat estigui entre 16 i 21 anys, s'aplicarà un descompte del 10%. En el cas que l'edat sigui més gran o igual que 65, el descompte serà del 30%. En la resta de casos, s'aplicarà la quota base.

**Nota:** Els valors dels paràmetres d'entrada sempre seran valors més grans que 0.

```
def calcular_quota(quota_base,edat):  
    if edat < 16:  
        return quota_base*0.8  
    elif edat < 21:  
        return quota_base*0.9  
    elif edat >= 65:  
        return quota_base*0.7  
    else:  
        return quota_base
```

#### Exercici 5 (1 punt)

Fer una funció anomenada `categories_socis` que rebí una llista d'anys de naixement de socis i l'any actual. La funció crearà i retornarà una nova llista on cada soci estarà classificat en un tipus de soci (1,2,3 ó 4) segons les següents categories d'edat:

Edat	Tipus
0 a 15 (infantil)	1
16 a 21(juvenil)	2
22 a 64 (adult)	3
>= 65 (senior)	4

**Exemple:** Si la funció rep la llista [2000,1950,2010,2003] i l'any actual és 2020, la llista retornada serà [2,4,1,2].

```
def categories_socis(llista_anys,any_actual):  
    llista_codis=[]  
    for x in llista_anys:  
        if any_actual-x < 16:  
            llista_codis.append(1)  
        elif any_actual-x < 22:  
            llista_codis.append(2)  
        elif any_actual-x < 65:  
            llista_codis.append(3)  
        else:  
            llista_codis.append(4)  
    return llista_codis
```

### Exercici 6 (1 punt)

Fer una **funció** anomenada `crear_llista` que permeti crear una llista d'una determinada dimensió i inicialitzada tota a un valor. La funció rebrà dos valors enters: el primer serà la dimensió de la llista i el segon el valor al qual s'hauran d'inicialitzar tots els elements de la llista. La funció retornarà la llista creada.

```
def crear_llista(dimensio,valor):  
    llista=[]  
    for i in range(dimensio):  
        llista.append(valor)  
    return llista
```

### Exercici 7 (1 punt)

Fer una funció anomenada `histograma` per conèixer la freqüència dels valors que es troben en una llista d'enters. La funció rebrà tres paràmetres:

- `L`, una llista d'enters
- `valor_min` i `valor_max`, dos nombres que indiquen el valor mínim i màxim de la llista

La funció haurà de retornar una altra llista on ens doni la freqüència d'aparició (número de vegades que apareix) dels valors entre `valor_min` i `valor_max` en la llista d'entrada.

**Nota:** Utilitzar la funció `crear_llista` de l'exercici 6 per crear la llista inicialitzada a zero.

```
def histograma(L,valor_min,valor_max):  
    frecuencia=crear_llista(valor_max-valor_min+1,0)  
    for x in L:  
        frecuencia[x-valor_min]+=1  
    return frecuencia
```

## Exercici 8 (2 punts)

Fer un programa que segueixi els següents passos:

1. Inicialitzacions de variables i constants.
2. Utilitzar (NO IMPLEMENTAR) el procediment `menu_principal()` que permet imprimir per pantalla el següent menú:

```
----- MENU -----  
1.- Donar d'alta soci  
2.- Calcular quota  
3.- Estadístiques  
4.- Finalitzar
```

3. Demanar a l'usuari que introdueixi una de les opcions.
4. Si l'opció és 1,
  - 4.1. Es demanarà que s'introdueixi el DNI del nou soci. Es comprovarà que el DNI sigui correcte cridant a la funció de l'ex. 2, `dni_correcte`. Si el DNI no és correcte escriurem el següent missatge d'error:

Error: DNI incorrecte

i tornarem a demanar el DNI. El programa no avançarà fins que s'introdueixi un DNI correcte.

- 4.2. Un cop el DNI sigui correcte es cridarà a la funció de l'ex. 3 `afegir_element` per inserir el DNI a la llista de dni's. En el cas que no es pugui inserir, informarà a l'usuari amb el següent missatge:

Error: DNI duplicat

Si s'ha pogut inserir, es demanarà l'any de naixement del soci i s'afegirà a la llista d'anys de naixement dels socis. I a continuació es mostrarà:

El soci s'ha donat d'alta correctament

5. Si l'opció és 2, s'imprimirà per pantalla el DNI i la quota a pagar per tots els socis (una línia per soci). Per calcular l'import a pagar, utilitzeu la funció del ex. 4 `calcular_quota`. La quota base per aquest any és de 90€.
6. Si l'opció és 3,
  - 6.1. Cridar la funció de l'ex. 5 `categories_socis` per crear la llista de les categories (1, 2, 3 ó 4) de tots els socis.
  - 6.2. Cridar la funció de l'ex. 7 `histograma` per conèixer la distribució de socis per categories i imprimir aquesta informació amb missatges tipus:

Categoria X - N socis

on X serà una de les categories (1, 2, 3 ó 4) i N el número de socis d'aquella categoria.

7. Si l'opció és 4, mostrar el missatge: `Sortint del programa...`
8. Qualsevol altra opció, escriure el missatge: `Opció no permesa.`
9. Repetir els passos 2 a 8, fins que l'opció del menú escollida sigui la 4.

```

#PROGRAMA PRINCIPAL
QUOTA_BASE=90
ANY_ACTUAL=2020
llista_dnis=[]
llista_anys=[]
opcio=0

while opcio!='4':
    menu_principal()
    opcio=input("Introdueix una opció: ")
    if opcio == '1':
        dni_soci=input("Introdueix el dni del soci: ")
        while not(dni_correccte(dni_soci)):
            print("Error: DNI incorrecte")
            dni_soci=input("Introdueix el dni del soci: ")
        if afegir_element(llista_dnis,dni_soci):
            any_soci=int(input("Introdueix l'any de naixement del soci: "))
            llista_anys.append(any_soci)
            print("El soci s'ha donat d'alta correctament")
        else:
            print("Error: DNI duplicat")
    elif opcio == '2':
        for i in range(len(llista_dnis)):
            print(llista_dnis[i],"-",calcular_quota(QUOTA_BASE,ANY_ACTUAL-
llista_anys[i]))
    elif opcio == '3':
        categories=categories_socis(llista_anys,ANY_ACTUAL)
        frequencies=histograma(categories,1,4)
        for i in range(4):
            print("Categoria",i+1,"-",frequencies[i],"socis")
    elif opcio == '4':
        print("Sortint del programa...")
    else:
        print("Error: Opció no permesa")

```

### Exercici 9 (1 punt) - versió 1

Si l'usuari introdueix valors a=2, b=4 i c=8, quins seran els valors de a, b i c al finalitzar el programa?

```
def acumular(x,y,L):
    llista_2=llista.copy()
    suma=x
    for i in range(len(L)):
        suma+=y
        L[i]=suma
    return llista_2

def shuffle(x,y,llista):
    llista_2=llista.copy()
    i=0
    for x in llista:
        llista_2[i]=x+i
        i+=1
    return llista_2

def bifurcar(a,b,llista):
    if a<b:
        res=acumular(a,b,llista)
        b=0
    else:
        res=shuffle(b,a,llista)
        a=0
    return a,res[1]

a=int(input("Introdueix el valor de a: "))
b=int(input("Introdueix el valor de b: "))
c=int(input("Introdueix el valor de c: "))
llista=[c,c,c]
a,b=bifurcar(a,b,llista)
c=llista[2]
print(a,b,c)
```

Valor a	Valor b	Valor c
2	8	14

### Exercici 9 (1 punt) - versió 2

Si l'usuari introdueix valors a=4, b=2 i c=8, quins seran els valors de a, b i c al finalitzar el programa?

```
def acumular(x,y,L):
    llista_2=llista.copy()
    suma=x
    for i in range(len(L)):
        suma+=y
        L[i]=suma
    return llista_2

def shuffle(x,y,llista):
    llista_2=llista.copy()
    i=0
    for x in llista:
        llista_2[i]=x+i
        i+=1
    return llista_2

def bifurcar(a,b,llista):
    if a<b:
        res=acumular(a,b,llista)
        b=0
    else:
        res=shuffle(b,a,llista)
        a=0
    return a,res[1]

a=int(input("Introdueix el valor de a: "))
b=int(input("Introdueix el valor de b: "))
c=int(input("Introdueix el valor de c: "))
llista=[c,c,c]
a,b=bifurcar(a,b,llista)
c=llista[2]
print(a,b,c)
```

Valor a	Valor b	Valor c
0	9	8