Departament de Disseny i Programació de Sistemes Electrònics
UNIVERSITAT POLITÈCNICA DE CATALUNYA

EXERCICI PUNTUABLE TECPRO

24/04/2017 Grau en Enginyeria de Sistemes TIC

COGNOMS:

NOM: GRUP de LAB:

Exercici 1. Relacions entre classes i excepcions. Donada la següent definició de classes, [**Apartat a**] justifiqueu què s'escriu per pantalla després d'executar el main.

```
class Unknown(object):
    x = 0
    y = 10
    def __init__(self, y=0):
        self.x = y
    def a1(self, y):
        self.x = min(self.x, y)
        return self.x
    def a2(self,y=0):
        self.x = min(self.x, self.y)
        return self.x+y
    def __str__(self):
        return type(self).__name__+" "+str(self.x)
    def __eq__(self,other):
        return self.x==other.x

class Important(Unknown):
    def suma(self,y=0):
        self.y = self.x + self.x
        return self.y+y
```

```
if __name__=='__main__':
    f=Unknown(5)
    print f #ex 1.1
    print f.a1(7) #ex 1.2
    g=Important(3)
    print f.a1(3) #ex 1.3
    print g.a1(5) #ex 1.4
    print g.suma(1) #ex 1.5
    print f.a2() #ex 1.6
    print g.a2() #ex 1.7
    g=Important(0)
    print f.suma() #ex 1.8
```

[**Apartat b**] En la sentència *g=Important(3)* a quin mètode es crida? Justifica si aquest mètode és un exemple de sobrecàrrega/redefinició/herència/delegació de mètodes. [**Apartat c**] En la sentència *g.a2()* a quin mètode es crida? Justifica si aquest mètode és un exemple de sobrecàrrega/redefinició/herència/delegació de mètodes. [**Apartat d**] Ara afegim la següent definició de classe contenidora elsUnknown, i se us demana justificar què s'escriurà per pantalla en executar el main.

```
class elsUnknown(object):
    def __init__(self):
        self.dades={}

    def add(self,dada):
        if dada not in self.dades:
            self.dades[dada]=repr(dada)
        else:
            raise Exception("Duplicada")
    def __str__(self):
        return "Els unknown info "+str(len(self.dades))+"\n"+",".join([str(element) for element in self.dades])

if __name__=='__main__':
    k=elsUnknown()
    r=Unknown()
    j=Important(2)
    try:
        k.add(r)
        k.add(j)
        k.add(r)
    except Exception as e:
        print e
    print k
```

Exercici 2. [**Apartat a**] Escriu la funció **recursiva**, de nom *ordenaD*, tal que, donada una llista no ordenada, *laLlista*, retorni la llista resultant d'ordenar-la descendentment, seguint l'estratègia que es detalla a continuació. L'element més gran anirà a la primera posició de la nova llista, i després cal realitzar l'ordenació amb la resta de la llista, de manera successiva fins que tota ella estigui ordenada. Podeu utilitzar els mètodes *len* i *remove* sobre llistes, i també el mètode max (l'ús de qualsevol altre mètode predefinit de Python o la implementació d'una funció no recursiva comportarà puntuació nul·la). [**Apartat b**] Escriviu el resultat dels doctest que segueixen, i afegeix un docstring.

```
def nose(x,param=0,div=0):
    """
    >>> nose(8)
    #3.1
    >>> nose(7)
    #3.2
    """
    if div>=3:
        return False
    else:
        if x-param!=0:
            if x%(x-param)==0:
                div+=1
                param+=1
                return nose(x,param,div)
            else:
                param+=1
                return nose(x,param,div)
        else:
            return True
```

Exercici 3. Donat el diagrama UML corresponent a la xarxa social iTICInsta que segueix, creeu l'esquelet de les classes, on es vegi clarament la materialització de cadascuna de les relacions entre classes, incloent el mètode constructor de cadascuna d'elles.