



**Universidad del Desarrollo**  
Facultad de Ingeniería

**Tecnologías de Información 2**  
**Examen**

**4 de diciembre de 2018**

**Instrucciones:**

- Indique su nombre y sección.
- Lea atentamente el enunciado de cada uno de los problemas
- Escriba con letra legible la respuesta.
- Duración: 2 horas

**Nombre:**

**Sección:**

**(2 puntos)** Problema 1

Programe el tipo de dato Reloj para que presente tiempo en un reloj de 24 horas. Por ejemplo, su tipo de dato debe ser capaz de representar 00:00, 13:30 o las 23:50.  
El tiempo es medido en horas (00 - 23) y minutos (00 - 59).

API para tipo de dato Reloj		
	Reloj(h, m)	<b>(0.5pts)</b> Crea un tipo de dato Reloj cuya tiempo es h horas y m minutos. Asuma h esté en el rango [0,23] y que m está en el rango [0, 59].
str	texto()	<b>(0.5pts)</b> Retorna una representación en string de este reloj. El formato que debe retornar es HH:MM (dos dígitos para la hora, y dos dígitos para minutos). Si la hora o minuto tiene sólo un dígito, debe anteponer un cero.
bool	esMasTemprano(b)	<b>(0.5pts)</b> Retorna True si el reloj actual está más temprano que el reloj b. Debe retornar False en otro caso. El parámetro b debe ser de tipo Reloj.
-	tic()	<b>(0.5pts)</b> Debe añadir un minuto al reloj actual.

Asuma que probaremos su solución utilizando el siguiente código:

```
r = Reloj(9, 59)
print(r.texto()) #debería imprimir '09:59'
r.tic()
print(r.texto()) #debería imprimir '10:00'
b = Reloj(11, 20)
print(r.esMasTemprano(b)) #debería imprimir True
```

Escriba su respuesta en el recuadro de la siguiente página.

### Respuesta Problema 1:

```
class Reloj:
    def __init__(self, h, m):
        self.h = h
        self.m = m
    def texto(self):
        r = ''
        if self.h <= 9:
            r += '0'
        r += str(self.h)
        r += ':'
        if self.m <= 9:
            r += '0'
        r += str(self.m)
        return r
        #solucion alternativa
        #return '{:02d}:{:02d}'.format(self.h, self.m)

    def tic(self):
        self.m += 1
        if self.m == 60:
            self.m = 0
            self.h += 1
        if self.h == 24:
            self.h = 0

    def esMasTemprano(self, b):
        if self.h < b.h:
            return True
        elif self.h == b.h and self.m < b.m:
            return True
        else:
            return False
        # solución alternativa
        #if self.h*60 + self.m < b.h*60 + b.m: return True
        #return False
```

**(2 puntos)** Problema 2

Analice el siguiente código y luego responda:

- a. **(1 pto.)** ¿Qué realiza la función f(s)? Demuéstrelo haciendo la traza del ciclo for en la línea 4 y 5 para s= 'aacgt'.
- b. **(1 pto.)** ¿Qué imprime el código? Hint: revise el ciclo for de las líneas 19 y 20.

```
1 def f(s):
2     n = len(s)
3     r = ''
4     for i in range(n):
5         r += s[n-i-1]
6     return r
7
8 L = ['aacgt', 'tgcaa', 'aacgt', 'aaca', 'aaca']
9 d = dict()
10 for b in L:
11     r = f(b)
12     if b in d:
13         d[b] += 1
14     elif r in d:
15         d[r] += 1
16     else:
17         d[b] = 1
18
19 for k,v in d.items():
20     print(k, ': ', v)
```

Complete la traza en el cuadro de la siguiente página.

## Respuesta Problema 2

a) ¿Qué hace la función  $f(s)$ ? **Entrega el string s en orden reverso.** (0.2 pts)

Escriba la traza de la función  $f('aacgt')$  en el siguiente recuadro: (0.8 pts)

Línea	s	n	r	i
2	aacgt	5	-	-
3	aacgt	5	' '	-
4	aacgt	5	' '	0
5	aacgt	5	't'	0
4	aacgt	5	't'	1
5	aacgt	5	'tg'	1
4	aacgt	5	'tgc'	2
5	aacgt	5	'tgc'	2
4	aacgt	5	'tgca'	3
5	aacgt	5	'tgca'	3
4	aacgt	5	'tgcaa'	4
5	aacgt	5	'tgcaa'	4
6	aacgt	5	'tgcaa'	4

b) ¿Qué imprime el código? Complete su respuesta en el siguiente cuadro:  
**(0.5 pts si tiene correcta las claves, 0.5 si tiene correcto los valores)**

aacgt : 3  
aaca : 2

**(2 puntos) Problema 3**

La función `impar()` recibe tres variables binarias (True o False) y retorna True si un número impar (1 o 3) son verdaderas; de otro modo, retorna False. Complete dos implementaciones de la función `impar` llenando la letra correspondiente a la expresión en los espacios correspondientes. Puede usar la misma letra una vez, más de una vez o ninguna vez. No puede usar código extra.

- |          |          |                  |               |
|----------|----------|------------------|---------------|
| A. True  | C. x     | G. x and y       | K. count += 1 |
| B. False | D. y     | H. x or y        |               |
|          | E. z     | I. x and y and z |               |
|          | F. not z | J. x or y or z   |               |

a) (1 pto. por cada letra correcta, % de punto)

```
def impar(x, y, z):  
    count = 0  
    if C: K  
    if D: K  
    if E: K  
    return (count%2) != 0
```

b) (1 pto. por cada letra correcta, % de punto)

```
def impar(x, y, z):  
    if G: return E  
    elif H: return F  
    else: return E
```

**(1 pto.)** Bonus: resolver sistema de ecuaciones lineales usando biblioteca numpy.

Transforme el siguiente sistema de ecuaciones lineales

$$5x + y = 13$$

$$3x + 2y = 5$$

en una ecuación de la forma  $Ax = b$ , donde A corresponde a la matriz formada por los coeficientes que acompañan a  $x$  e  $y$ ; y  $b$  es un vector con los coeficientes al lado derecho de la igualdad.

Luego resuelva la ecuación calculando  $x = A^{-1}b$  con el módulo numpy.

Recuerde que `np.linalg.inv(A)` retorna la inversa de la matriz A.

Complete su solución en el siguiente cuadro:

**(0.25 puntos por línea correcta)**

```
1 import numpy as np
2 A = np.array([[5, 1],[3,2]])
3 b = np.array([13, 5])
4 Ainv = np.linalg.inv(A)
5 x = Ainv.dot(b)
6 print(x) #opcional
7
8
9
10
11
12
13
14
15
16
17
18
19
20
```