

---

NOMBRE Y APELLIDOS:

---

1. [15 %] ORDENADORES Y CODIFICACIÓN

1. Si el reloj de la CPU de un computador es de 2.5GHz, y se requiere ejecutar un programa de Python de 5 instrucciones ¿Cuánto tiempo tardará en ejecutarse?

2. Se consideran los números en binario  $N = 100011011$  y  $M = 1101011$ .

- Calcular en binario la suma y la resta de  $N + M$  y  $N - M$
  
  
  
  
  
  
  
  
  
  
- Calcular en binario la división euclídea (cociente y resto) de  $N$  entre  $M$ .
  
  
  
  
  
  
  
  
  
  
- Convertir a octal  $N$  y a hexadecimal  $M$ .

3. Escribe en notación hexadecimal la codificación UTF-8 del carácter cuyo *code point* UCS en hexadecimal es  $U + 950$ .

2. [10 %] BOOLEANOS

Se consideran el `prog1.py` utilizando una única bifurcación condicional:

```
_____ prog1.py _____  
1 b=float(input(" Dame un numero: "))  
2 a=int(input(" Dame un numero entero: "))  
3  
4 if (a%7>0 and not b==1) and (1<=b>a or a*(b-1) !=1):  
5     print(1)  
6 else:  
7     print(0)
```

```
_____ prog2.py _____  
1 b=float(input(" Dame un numero: "))  
2 a=int(input(" Dame un numero entero: "))  
3  
4 if ## CONDICIÓN  
5     print(0)  
6 else:  
7     print(1)
```

Completar el programa `prog2.py` para que tenga el mismo resultado que `prog1.py`, pero escribiendo la condición contraria a la que se utiliza en `prog1.py`, y sin utilizar el operador `not`.

#ESCRIBIR AQUI LA CONDICION:

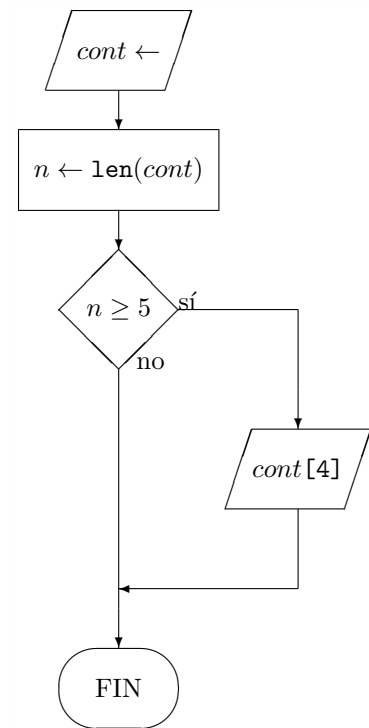
3. [15 %] DIAGRAMAS DE FLUJO

Escribe un diagrama de flujo que represente el programa escrito abajo, a la izquierda, y el código Python correspondiente al diagrama de flujo de la derecha. ¿Tienen el mismo resultado final estos dos programas? En caso contrario, ¿en qué se diferencian?

```

1 cont = lista
2 n = len(cont)
3 i = 0
4 while i < 4 and i < n - 1:
5     i += 1
6 print(cont[i])

```



4. [15 %] BUCLES

¿ Qué se muestra por pantalla después de ejecutar estos dos programas ?

```

##PROGRAMA 1
j = 5
for i in range(2):
    while j > 2:
        print('{0}, {1}'.format(j, i), end='+')
        j -= 1

```

```

##PROGRAMA 2
j = 5
while j > 2:
    for i in range(2):
        print('{1}, {0}'.format(i, j), end='+')
        j -= 2

```

SOLUCION(PROGRAMA 1):

SOLUCION(PROGRAMA 2):

## 5. [15 %] LISTAS

Se quiere modificar una lista, desplazando sus valores una posición a la derecha, tal que el último elemento sea el primero.

```
>>> desplaza_derecha ([5, 10, 15, 20, 25, 30, 35])
[35, 5, 10, 15, 20, 25, 30]
```

- Completa los puntos suspensivos y corrige la función siguiente para lograrlo (no usar ninguna otra lista que no sea v).

```
def desplaza_derecha(v):
    aux=v[-1]
    for i in range(len(v)-1,-1,-1):
        v[i+1]=v[i]
    ...

    return v
```

- Análogamente, escribe una función que desplace los elementos de una lista cualquiera, una posición hacia la izquierda.

```
>>> desplaza_izda ([5, 10, 15, 20, 25, 30, 35])
[10, 15, 20, 25, 30, 35, 5]
```

SOLUCION:

```
def desplaza_izda(v):
```

## 6. [10 %] DICCIONARIOS

Escribir qué se muestra por pantalla cuando se ejecuta el siguiente programa:

```
D1 = {'a' : 'uno', 'b' : 'dos', 'c' : 'tres', 'd' : 'cuatro'}
D2 = {'c' : 'cinco', 'd' : 'seis', 'e' : 'siete', 'f' : 'ocho'}
D = { }
for i in D1:
    D[i] = D1[i]
for d in D2:
    D[d] = D2[d]
for d in D:
    print(d, D[d], end =',')
```

SOLUCION:

## 7. [20%] VACACIONES EN LA ESTACIÓN DE ESQUÍ Y MONTAÑA ALTO CAMPOO

*Sveta Campoo Bank* ha lanzado, recientemente, un préstamo con condiciones generosas para disfrutar de unas merecidas vacaciones este próximo invierno, en la esperanza de haber exterminado el COVID-19.

Se pide escribir una función `sveta_campoo_plan` mostrando el plan de pago del banco por una cantidad prestada. La cantidad se divide en varias cuotas mensuales. Después del primer pago, se asigna una comisión. La comisión es un porcentaje de la cantidad de la cuota y aumenta cada mes: sin comisión en el primer mes, 10% de la cuota en el segundo mes, 20% de la cuota en el tercer mes y así sucesivamente. Entonces, la función `sveta_campoo_plan` debe tener dos argumentos, la cantidad adeudada/prestada y el número de cuotas mensuales; deberá mostrar por pantalla: la cantidad a pagar correspondiente a cada mes y la comisión total.

```
>>> sveta_campoo_plan(1000,1)
Cuota = 1000.0
#1: 1000.0
Comision: 0.0
>>> sveta_campoo_plan(1000,4)
Cuota = 250.0
#1: 250.0
#2: 275.0
#3: 300.0
#4: 325.0
Comision: 150.0
```

```
>>> sveta_campoo_plan(1000,2)
Cuota = 500.0
#1: 500.0
#2: 550.0
Comision: 50.0
>>> sveta_campoo_plan(1000,3)
Cuota = 333.33
#1: 333.33
#2: 366.67
#3: 400.0
Comision: 100.0
```

SOLUCION:

```
def sveta_campoo_plan(deuda, cuotas):
```