



NOMBRE Y APELLIDOS:

1. [2%] ¿Qué resultará de ejecutar las siguientes sentencias ?

```
>>> z = 2
>>> z += 2
>>> z += 2 - 2
>>> z *= 2
>>> z *= 1+1
>>> z //= 2
>>> z %= 3
>>> z /= 3 -1
>>> z -= 2 + 1
>>> z -= 2
>>> z **= 3
>>> z
```

<----- Solucion

2. [20%] Se considera el siguiente código de Python:

```
j = -1; Lista=[]
for i in range(2, 5):
    Lista = Lista + [i+j] + Lista
    j = j + 1
```

1. ¿Cuál es el valor de i , j y $Lista$, después de ejecutarlo ?

SOLUCION:

2. Escribir el código usando `while` en lugar de `for in ...`

SOLUCION:

3. [18%] Transformar a binario (usando a lo sumo 16 bits) el número decimal 126.12. Transformar a octal y a hexadecimal el número binario obtenido.

BINARIO

OCTAL

HEXADECIMAL

4. [20%] Escribe un programa para encontrar (si existe) una solución entera positiva del sistema de ecuaciones

$$2x + y = A$$

$$6x + 3y = B$$

y donde A y B son dos enteros leídos por teclado.

AYUDA: Usar un bucle para iterar todas las posibles soluciones.

NOTA: Puede que no tenga solución para algunos pares A y B .

>>>

A = 4

B = 12

Solución x=1, y=2.

>>>

A = 6

B = 4

No hay solución

=====

5. [20 %] Utiliza la sentencia `continue` para enumerar los cinco primeros años bisiestos que siguen a una fecha dada por el usuario.

```
$ python bisiestos.py
```

```
Año: 2014
```

```
1: 2016
```

```
2: 2020
```

```
3: 2024
```

```
4: 2028
```

```
5: 2032
```

6. [20 %] ¿Qué resultará de ejecutar las siguientes expresiones en PYTHON ? Algunos de ellas pueden producir errores de teclado y/o excepciones.

- `+ - - - 2`

- `1 + 2 * *2 * *3`

- `1.0 + 3 %2`

- `1/2.0/4`

- `1//2/4`

- `4**.5`

- `4.0**(1//2)`

- `10//5E-3`

- `1j**2`

- `True == True != 1`

- `type(True + 2)`

- `1 or abcd`

- 0 or abcd
- (1 != 2 <5) and (10 <23)
- "fresno" <= "santander"
- 0 and abcd
- print(2**0 * False + 2**2**True * False+2**True,"Torre", "la", False)
- "hola"[0][0]
- 'hola'[-1][0]
- str(12)
- tuple(12)
- type(['silencio',1,[5.0,5]][0][2]);
- 'a'*3+'/'**5+2*'abc'+'+'
- len([1,[]])
- float([3])
- (1,2,3,4,5,6,7)[1:5:3]
- bool('hola')
- [1, 2, 3, 4, 5, 6, 7][-5 : -2]
- 'hola santander'[1:5]
- type([1,2,3,4,5][2:4:3])
- type((1))
- type((1,))
- a =[1,2]; a[0] is 1.0
- 'ol' in "hola, Santander"
- [1] in [1,2,3]
- (1,2) in (1,2,3)
- bool(abcd)
- bool(3)