

1. Iniciar una sesión de trabajo en GNU-Linux.
2. Muestre el árbol de directorios de su directorio HOME.
3. Sitúese en el directorio de la asignatura "Programación Científica" (`cd PC`).
4. Muestre el contenido del directorio de trabajo (`ls -la`).
5. Cree un nuevo directorio denominado *prct02* (`mkdir prct02`).
6. Sitúese en el directorio *prct02* (`cd prct02`) y cree la estructura de directorios que le permita tener subcarpetas para el código y los documentos, es decir:
 - un subdirectorio *src*
 - un subdirectorio *docs*
7. Guarde el fichero PDF que contiene el enunciado de esta práctica en el directorio *docs*.
8. En el directorio *docs* cree un fichero `respuestas.txt` en que almacene las respuestas a las preguntas de los ejercicios.
9. Ejecutar el intérprete interactivo de Python (`python`). Recuerde que para salir el intérprete interactivo se ha de teclear un carácter de final de fichero (`Control+D`).
10. ¿Qué hace el siguiente programa? ¿Es correcto? ¿Qué está mal?

```
a = float(raw_input('Valor de a: '))
b = float(raw_input('Valor de b: '))

a * x + b = 0

print 'Solucion: ', x
```

11. ¿Qué hace el siguiente programa? ¿Es correcto? ¿Qué está mal?

```
x = -b/a

a = float(raw_input('Valor de a: '))
b = float(raw_input('Valor de b: '))

print 'Solucion: ', x
```

12. ¿Qué hace el siguiente programa cuando el valor de la variable **a** es cero? Haga una propuesta para solucionar el error que se produce en tiempo de ejecución.

```
a = float(raw_input('Valor de a: '))
b = float(raw_input('Valor de b: '))

x = -b/a

print 'Solucion: ', x
```

13. ¿Qué error se produce en tiempo de compilación?. ¿Cómo se soluciona?

```
a = float(raw_input('Valor de a: '))
b = float(raw_input('Valor de b: '))

if a != 0:
    x = -b/a
    print 'Solucion: ', x
if a = 0:
    print 'La ecuación no tiene solución.'
```

14. Modifique el programa que calcula la solución de una ecuación de primer grado para que contemple que cuando tanto el valor de a como el de b son cero, la ecuación tiene infinitas soluciones.

15. ¿Qué hace el siguiente programa?

```
from math import sqrt

a = float(raw_input('Valor de a: '))
b = float(raw_input('Valor de b: '))
c = float(raw_input('Valor de c: '))

if a != 0:
    x1 = (-b + sqrt(b**2 - 4*a*c)) / (2 * a)
    x2 = (-b - sqrt(b**2 - 4*a*c)) / (2 * a)
    print 'Las soluciones de la ecuacion son: x1=%4.3f y x2=%4.3f' % (x1, x2)
else:
    if b != 0:
        x = -c / b
        print 'La solucion de la ecuacion es: x=%4.3f' % x
    else:
        if c != 0:
            print 'La ecuacion no tiene solucion'
        else:
            print 'La ecuacion tiene infinitas soluciones'
```

16. ¿Existe alguna diferencia entre el programa de la pregunta 15 y este cuando se ejecutan?

```
from math import sqrt

a = float(raw_input('Valor de a: '))
b = float(raw_input('Valor de b: '))
c = float(raw_input('Valor de c: '))

if a == 0:
    if b == 0:
        if c == 0:
            print 'La ecuacion no tiene solucion'
        else:
            print 'La ecuacion tiene infinitas soluciones'
    else:
        x = -c / b
        print 'La solucion de la ecuacion es: x=%4.3f' % x
else:
    x1 = (-b + sqrt(b**2 - 4*a*c)) / (2 * a)
    x2 = (-b - sqrt(b**2 - 4*a*c)) / (2 * a)
    print 'Las soluciones de la ecuacion son: x1=%4.3f y x2=%4.3f' % (x1, x2)
```

17. ¿Existe alguna diferencia entre el programa de la pregunta 15 y este cuando se ejecutan?

```
from math import sqrt

a = float(raw_input('Valor de a: '))
b = float(raw_input('Valor de b: '))
c = float(raw_input('Valor de c: '))
```

```

if a == 0 and b == 0 and c == 0:
    print 'La ecuacion tiene infinitas soluciones'
else:
    if a == 0 and b == 0:
        print 'La ecuacion no tiene solucion'
    else:
        if a == 0:
            x = -c / b
            print 'La solucion de la ecuacion es: x=%4.3f' % x
        else:
            x1 = (-b + sqrt(b**2 - 4*a*c)) / (2 * a)
            x2 = (-b - sqrt(b**2 - 4*a*c)) / (2 * a)
            print 'Las soluciones de la ecuacion son: x1=%4.3f y x2=%4.3f' % (x1, x2)

```

18. Modifique el programa de la pregunta 17 para utilizar la sentencia `elif`.
19. ¿Qué ocurre cuando ejecuta el programa de la pregunta 17 con los siguientes valores $a = 1$, $b = 1$ y $c = 1$? Proponga una solución.
20. ¿Qué hace el siguiente programa?

```

sumatorio = 0
i = 1
while i <= 10:
    i += 1
    sumatorio += i
print sumatorio

```

21. ¿Cuál es la diferencia entre este programa y el de la pregunta 20? ¿Producen ambos el mismo resultado? ¿Por qué?

```

sumatorio = 0
i = 1
while i <= 10:
    sumatorio += i
    i += 1
print sumatorio

```

22. ¿Qué hace el siguiente programa?

```

numero = int( raw_input('Introduzca un numero '))

for potencia in [2,3,4,5]:
    print '%d elevado a %d es %d' % (numero, potencia, numero**potencia)

```

23. ¿Qué muestran las siguientes sentencias? ¿Cuál es el tipo de datos de la variable `a`?

```

>>> a = ['pan', 'huevos', 100, 1234]
>>> a
?
>>> a[0]
?
>>> a[3]
?
>>> a[-2]
?
>>> a[1:-1]
?
>>> a[:2] + ['carne', 2*2]
?
>>> 3*a[:3] + ['Boo!']

```

```

?
>>> a
?
>>> a[2] = a[2] + 23
>>> a
?
>>> a[0:2] = [1, 12]
>>> a
?
>>> len(a)
?
>>> q = [2, 3]
>>> p = [1, q, 4]
>>> len(p)
?
>>> p[1]
?
>>> p[1][0]
?
>>> p[1].append('extra')
>>> p
?
>>> q
?
```

24. ¿Qué hacen las siguientes sentencias?

```

>>> a = ['gato', 'ventana', 'defenestrado']
>>> for x in a:
>>>     print x, len(x)
>>>
>>>?
>>> for x in a[:]:
>>>     if len(x) > 6: a.insert(0,x)
>>>     print a
>>>
>>>?
```

25. ¿Qué hacen las siguientes sentencias?

```

>>> r = range(5,10)
>>> print r
>>> ?
>>> s = range(0, 10, 3)
>>> print s
>>> ?
>>> a = ['gato', 'ventana', 'defenestrado']
>>> for i in range(len(a)):
>>>     print i, a[i]
>>> ?
```

26. ¿Qué hace el siguiente programa?

```

for i in range(0,5):
    for j in range(0,3):
        print i, j
```

27. ¿Qué hace el siguiente programa?

```

for i in range(0,5):
    for j in range(i,5):
        print i, j
```

28. ¿Qué hace el siguiente programa?

```
for i in range(0,5):
    for j in range(0,i):
        print i, j
```

29. ¿Qué hace el siguiente programa?

```
for i in range(0,4):
    for j in range(0,4):
        for k in range(0,2):
            print i, j, k
```

30. ¿Qué hace el siguiente programa?

```
for i in range(0,4):
    for j in range(0,4):
        for k in range(i,j):
            print i, j, k
```

31. ¿Qué hace el siguiente programa?

```
for i in range(1,5):
    for j in range(0,10,i):
        print i, j
```

32. ¿Qué hay mal en la siguiente función?

```
def es_perfecto(n):
    for i in range(1,n):
        sumatorio = 0
        if n % i == 0:
            sumatorio += i
    return sumatorio == n
```

33. ¿Qué hace el siguiente programa?

```
def tabla_perfectos(m):
    for i in range(1, m+1):
        if es_perfecto(i):
            print i, 'es perfecto'

x = int( raw_input('Introduzca un numero ') )
tabla_perfectos(x)
```

34. Situado en el directorio de la asignatura, es decir, en el directorio \PC comprima las actividades de la práctica (`tar -zcvf prct02.tgz prct02/`).

35. Compruebe que se ha creado el fichero `prct02.tgz` correctamente en el directorio actual (`tar -ztvf prct02.tgz *`).

36. Suba el fichero `prct02.tgz` a la tarea habilitada en el campus virtual.

37. Cierre la sesión.