

1. Ara agafeu un fitxer Python que hagueu treballat a M3 i realitzeu el següent:

1. Creeu els tests usant la llibreria unittest

```
import unittest
import m5
class cuadranum(unittest.TestCase):
    def test_cas1(self):
        #caso 1: la lista de varios numeros [1,2,3,4,5]
        self.assertEqual(cuadranum.cuadranum([1,2,3,4,5]),1,2,3,4,5)
    def test_cas2(self):
        #caso 2: la lista con dos numeros [1,2]
        self.assertEqual(cuadranum.cuadranum([1,2]),1,2)
    def test_cas3(self):
        #caso 3: la lista con un numeros [1]
        self.assertEqual(cuadranum.cuadranum([1]),1)
unittest.main()
```

2. Creeu els tests usant la llibreria doctest

```
#!/usr/bin/env python
# -*- coding: utf-8 -*-
#
import doctest
def cuadranum(w):
    """
    Retorna la lista de numeros.
    >>> cuadranum([1,2,3,4,5])
    [ 1, 4, 9, 16, 25]
    """

    #crear lista resultado
    lista=[]
    for i in range(0,len(w)):
        lista.append(w[i]*w[i])
    return lista

import doctest
doctest.testmod(verbose=True)
```

### 3. Annexeu captures conforme ho heu provat

```
Trying:
  cuadranum([1,2,3,4,5])
Expecting:
  [ 1, 4, 9, 16, 25]
*****
File "doctestpython.py", line 8, in __main__.cuadranum
Failed example:
  cuadranum([1,2,3,4,5])
Expected:
  [ 1, 4, 9, 16, 25]
Got:
  [1, 4, 9, 16, 25]
1 items had no tests:
  __main__
*****
1 items had failures:
  1 of 1 in __main__.cuadranum
1 tests in 2 items.
0 passed and 1 failed.
***Test Failed*** 1 failures:
doctest
doctest.testmod(verbose=True)
20

-----
(program exited with code: 0)
Press return to continue
```

2. En el fitxer Travessa.java del repositori següent: <https://github.com/sergiperez/m5uf2proves> heu de crear les proves unitàries que facin falta per assegurar una cobertura del 100%.
3. Heu de trobar els errors, detectar-los amb el depurador i solucionar-los.
4. Heu de crear un document on:
  1. Hi ha el diagrama de flux i càlcul dels camins.
  2. Les captures del depurador que indiquin com heu solucionat els errors.



