

```
#!/usr/bin/env python3
# -*- coding: utf-8 -*-

#
# fichier: monome_tests.py
# version: 0.5.0
# auteur: Pascal CHAUVIN
# date: 2014/10/28
#
# (tous les symboles non internationaux sont volontairement omis)
#

import sys
sys.path.append('../rationnel_mod')

import monome

import rationnel

def test_unitaire_0(visible =False):
    print("*** monome: test_unitaire_0 ***")

    t = monome.monome()
    if visible:
        print(t)
        print(repr(t))

    ok = True
    return ok

def test_unitaire_1(visible =False):
    print("*** monome: test_unitaire_1 ***")

    t = monome.monome(rationnel.rationnel(5, -10), "lapin")
    if visible: print(t)

    t.fixer_coeff(rationnel.rationnel(4))
    if visible: print(t)

    t.fixer_coeff(rationnel.rationnel(-4))
    if visible: print(t)

    ok = True
    return ok

def test_unitaire_2(visible =False):
    print("*** monome: test_unitaire_2 ***")

    lapin = monome.monome(rationnel.rationnel(5), "lapin")
    if visible: print(lapin)

    poulet = monome.monome(rationnel.rationnel(5), "poulet")
    if visible: print(poulet)

    poulet.fixer_coeff(rationnel.rationnel(4))

    ok = (lapin < poulet)
    return ok

def test_unitaire_3(visible =False):
    print("*** monome: test_unitaire_3 ***")
```

```
xax = monome.monome(rationnel.rationnel(5), "xax")
if visible: print(xax)

axx = monome.monome(rationnel.rationnel(7), "axx")
if visible:
    print(xax)
    print(repr(xax))

ok = not (xax == axx) and \
    (axx.lire_indet() == xax.lire_indet())
return ok

def test_unitaire_4(visible =False):
    print("*** monome: test_unitaire_4 ***")

    mauvais = monome.monome(rationnel.rationnel(7, -1), "MAUVAIS=", False)
    if visible:
        print(mauvais)
        print(repr(mauvais))

    ok = (not mauvais.est_valide())
    return ok

def test_unitaire_5(visible =False):
    print("*** monome: test_unitaire_5 ***")

    a = monome.monome(rationnel.rationnel(1), "?x")
    if visible: print(a)

    ok = True
    return ok

def test_unitaire_6(visible =False):
    print("*** monome: test_unitaire_6 ***")

    xaxax = monome.monome(rationnel.rationnel(5), "xaxax")
    if visible:
        print(xaxax)
        print(xaxax.joli())

    axxxx = monome.monome(rationnel.rationnel(7), "axxxx")
    if visible:
        print(axxxx)
        print(axxxx.joli())

    ok = not (xaxax == axxxx)
    return ok

def test_unitaire_7(visible =False):
    print("*** monome: test_unitaire_7 ***")

    ok = True
    return ok

def test_unitaire_8(visible =False):
    print("*** monome: test_unitaire_8 ***")
```

```
ok = True
return ok
```

```
def test_unitaire_9(visible =False):
    print("*** monome: test_unitaire_9 ***")
```

```
ok = True
return ok
```

```
def test_unitaire_(visible =False):
    print("*** monome: test_unitaire_ ***")
```

```
ok = True
return ok
```

```
def tests_unitaires():
    return (
        test_unitaire_0() and \
        test_unitaire_1() and \
        test_unitaire_2() and \
        test_unitaire_3(True) and \
        test_unitaire_4() and \
        test_unitaire_5(True) and \
        test_unitaire_6(True) and \
        test_unitaire_7() and \
        test_unitaire_8() and \
        test_unitaire_9()
    )
```

```
if __name__ == "__main__":
    ok = tests_unitaires()
    if ok:
        print("*** monome: tests unitaires OK ***")
```