# Palindromes & Lychrel

### TP3, Boucles et Listes ##

1

```
def palindrome(l):

    return l == l[::-1]
```

```
palindrome = lamba l: l == l[::-1]
```

2

```
def float repr(n):
    num, mantissa = str(n).split('.')
    return list(num) + list(mantissa)
```

3

```
def NbrPal(N):
    count = 0
```

αβγδεζηθκλμνξοπρσ τυφχψω

## Exercice 1

1 Plus rapide

```
k = 0
primes = [ ]
while 7*k < 100:

    primes += [
    k += 1
print(primes)
```

Exercice 3

$\boxed{2}$ ~~$|u_n - \sqrt{5}| < 10^{-5}$~~

~~$\Leftrightarrow |u_n - \sqrt{5}|^2 < 10^{-10}$~~

~~$\Leftrightarrow |u_n^2 - 2u_n\sqrt{5} + 5| < 10^{-10}$~~

$\Leftrightarrow$

$u_n - \sqrt{2}|^2 < 10^{-10}$

$\Leftrightarrow |u_n^2 - 2u_n\sqrt{2} + 2| < 10^{-10}$

$\Leftrightarrow |u_n^2 - 2u_n\sqrt{2}| + 2 < 10^{-10}$

$\Leftrightarrow |u_n^2 - 2u_n\sqrt{2}| < 10^{-10} - 2$

$\Leftrightarrow |u_n^2 - 2||u_n\sqrt{2}| < 10^{-10} - 2$

$\Leftrightarrow |u_n^2 - 2| < \frac{10^{-10} - 2}{|u_n\sqrt{2}|}$

---

```
n = 0
while  abs(u(n)**2-2) < (10e-10 - 2)/abs(u(n) * sqrt(2)):
      n += 1
print(n)
```

5 Palindromes & Lychrel

```
def hauteur-palindromique(N):
    hateur = 0

    while not palindrome(str(N)):
        hauteur += 1
        N = N + int(reversed(str(N)))
    return hauteur
```

4

```
def make_palindrome(N):
    while not palindreme(str(N)):
```

4+5

```
def make_pal(N, max_iter.int = 1e9) → tuple[int, int]:
    """Returns the palndre from N with lowest height, Returned as 2nd arg"""
    height = 0
    while not palindrme(str(N)):
        height += 1
        N += int(str(N)[::-1])
        if height > max_iter:
            raise RecursionDepthError()
    return N, height
```