Informatique **CPGE**
2^{de} année Clemenceau

TP 2 : Voyage en Pythonerie Corrigé

Projet 1. *Approximation de π*

```
## Approximations de pi
## Denis Renault
## Derniere version le 21 octobre 2014
##############################################


## Importation des modules
##########################################
import math as m
import fractions as f
import csv

## Declaration des fonctions
##########################################
def u_n(n):
    '''renvoie le nieme terme de la suite Un'''
## un = f.Fraction(0, 1) mais le module fraction ralentit enormement le programme
    un = 0
    for k in range (n + 1):
##          un += f.Fraction(1, pow((k + 1), 2))
        un  += 1 /  pow((k + 1), 2)
    return (float(un))

def pi1(n) :
    '''renvoie une estimation de pi a partir de Un'''
    return ( pow(6 * float(u_n(n)), 0.5) )

def v_n(n):
    '''renvoie le nieme terme de la suite Vn'''
##    vn = f.Fraction(0, 1)
    vn = 0
    for k in range (n + 1):
##          vn += f.Fraction(1, pow((k + 1), 6))
        vn += 1 / pow((k + 1), 6)
    return ( float(vn) )

def pi2(n) :
    '''renvoie une estimation de pi a partir de Vn'''
    return ( pow( 945 * float(v_n(n)), (1 / 6) ) )

def w_n(n):
    '''renvoie le nieme terme de la suite Wn'''
##    wn = f.Fraction(0, 1)
    wn = 0
    for k in range (n + 1):
##          wn += f.Fraction(pow(-1, k), 2 * k + 1)
        wn += pow(-1, k) / (2 * k + 1)
    return ( float(wn) )

def pi3(n) :
    '''renvoie une estimation de pi a partir de Wn'''
    return (4 * w_n(n))

def r_n(n):
    '''renvoie le nieme terme de la suite Rn'''
    rn = 0
    for k in range (n + 1):
        rn += (m.factorial (4 * k) * (1103 + 26390 * k)) / \
              ((m.factorial (k) ** 4) * 396 ** (4 * k))
    return ( ( pow(8, 0.5) / 9801 ) * rn )
```

```
def pi4(n) :
    '''renvoie une estimation de pi a partir de Rn'''
    return ( 1 / r_n(n) )

def decimales_correctes(val):
    '''renvoie le nombre de decimales identiques a la valeur de pi fournie par
le module math'''
    n, i = 0, 2
    # n est le compteur de decimales correctes.
    # i demarre a 2 car les deux premiers caracteres sont 3 et .
    correct = True
    while correct and i < min(len (str(val)), len (str(m.pi))):
        # Si toutes les decimales correspondent, il faut arreter la boucle
        # quand on a termine de scruter les chiffres du nombre comportant
        # le moins de decimales
        n += 1
        if str(val)[i] != str(m.pi)[i]:
            correct = False
            n -= 1
        i += 1
    return(n) # retourne le nombre de decimales correctes apres la virgule

def affichage(fonction, valeurs):
    '''affiche la valeur d'une fonction d'approximation de pi pour une liste
de valeurs'''
    for val in valeurs:
        approx = fonction(val)
        print(str(fonction).split()[1], '(', float(val), ') =', approx, \
              'soit', decimales_correctes(approx), 'decimales correctes.')

## Declaration des constantes
##########################################
Fonctions_pi = [pi1, pi2, pi3, pi4]

## Recuperation du tableau de donnees
##########################################
with open ('Approximations_pi_eleve.csv', newline = '') as file :
    lire = csv.reader(file)
    tab_initial = []
    for ligne in lire :
        tab_initial.append(ligne)

## Construction du tableau de donnees rempli
##################################################
tab_donnees = tab_initial

val = [ [] for i in range(len(Fonctions_pi))] # memorise les valeurs de n
ligne = 2
while ligne <= 11 :
    print('ligne', ligne)   # pour patienter pendant le traitement...
    for col in range(1, len(tab_initial[ligne])) :
        print('col', col)   # idem
        index = (ligne + 1) // 3 - 1
        n = int( eval( tab_initial[ligne][col] ) )
        val[index].append(n)
        estim = Fonctions_pi[index](n)
        tab_donnees[ligne + 1][col] = estim
        tab_donnees[ligne + 2][col] = decimales_correctes(estim)
    ligne += 3

## Exportation du tableau de donnees en csv
################################################
with open ('Approximations_pi_rempli.csv', 'w', newline = '') as file :
    ecrire = csv.writer(file)
    for ligne in tab_donnees :
        ecrire.writerow(ligne)

## Il ne reste plus qu'a ouvrir le fichier 'Approximations_pi_rempli.csv'
## avec LibreOffice
```