```
import time
import numpy as np
import cmath
import math
from math import log,isqrt
import mpmath
from mpmath import ei,li
import scipy
from scipy.integrate import quad

class Premiers():
  def __init__(self,n):
    premier = np.full(n,True)
    premier[:2] = False
    for p in range(2,math.isqrt(n)+1):
      if premier[p]:
        premier[p*p::p] = False
    self.__premiers = np.nonzero(premier)[0]
  def compte(self,x):
    return np.searchsorted(self.__premiers,x,side='right')

def f(t):
    return(1/((t-1)*t*(t+1)*log(t)))

def integraleaajouter(x):
    return(quad(f,x,np.inf)[0])

def expoi(x):
    fic = open("zeros10000",'r')
    zeros = fic.readlines()
    zeros = map(float,zeros)
    sommedesEi = 0
    for zz in zeros:
        sommedesEi = sommedesEi+ei((0.5+zz*1j)*log(x))+ei((0.5-zz*1j)*log(x))
    fic.close()
    return(sommedesEi)

tic = time.time()
P = Premiers(10001)
for x in range(2,10001):
    a = P.compte([x])
    print('pix(',x,') calcul par pgm --> ',int(a))
    formuleR = li(x)-expoi(x)+log(0.5)+integraleaajouter(x)-log(2)
    print('pix(',x,') calcul par formule --> ',formuleR.real)
    erreur = (int(formuleR.real)-a)/a
    print('erreur entre valeur reelle et valeur par formule
ei :',erreur[0],'\n')
```