```
# -*- coding: utf-8 -*-
"ce code est deja expliquer sur le wiki"

import numpy
import random
import math

class Ising2D:
    def __init__(self,N):
        self.N = N
        self.Ns = N*N
        self.spin = numpy.zeros((N,N),dtype=numpy.uint8)
        self.nbits = 8
        self.Nspin = self.Ns*self.nbits

    def temperature(self,T):
        beta = 1.0/T
        p0 = math.exp(-8.0*beta)
        a=math.exp(-4.0*beta)
        b=a+p0
        p1 = a/(b)
        self.deuxp0 = p0
        self.deuxp1 = p1

    def voisin(self,i,j):
        ii = i
        if ii<0:
            ii=self.N-1
        elif ii>=self.N:
            ii=0
        jj = j
        if jj<0:
            jj=self.N-1
        elif jj>=self.N:
            jj=0
        return self.spin[jj][ii]


    def metropolis(self):
        i = random.randint(0,self.N-1)
        j = random.randint(0,self.N-1)
        s = self.spin[j][i]
        a1 = s^self.voisin(i-1,j)
        a2 = s^self.voisin(i+1,j)
        a3 = s^self.voisin(i,j-1)
        a4 = s^self.voisin(i,j+1)
        R1 = a1|a2|a3|a4
        R2 = ((a1|a2)&(a3|a4))|((a1&a2)|(a3&a4))
        if random.random() < self.deuxp0:
            r0 = random.getrandbits(self.nbits)
        else:
            r0 = 0
    if random.random() < self.deuxp1:
        r1 = random.getrandbits(self.nbits)
    else:
        r1 = 0
    self.spin[j][i] ^= R2|(R1&r1)|r0



def moment(self):
    m = 0.0
    for i in range(self.N):
        for j in range(self.N):
            s = self.spin[j][i]
            for b in range(self.nbits):
                m += 2*(s&1)-1
                s = s >> 1
    return -m*1.0/self.Nspin

def boucle(self,n):
    m = numpy.zeros(n)
    for k in range(n):
        m[k] = self.moment()
        for i in range(self.Ns):
            self.metropolis()
    return (m,numpy.mean(m),numpy.std(m))

def couche(self):
    mask = numpy.ones((self.N,self.N),dtype=numpy.uint8)
    return numpy.bitwise_and(self.spin,mask)
```