```
"""
Registration: 012-1111-0461-20;
Roll: 203012-21-0008
Description: Orthonormality and recursion relation for Chebyshev functions
Author: Chitrak Roychowdhury
"""


import numpy as np
from scipy.special import chebyt
from scipy.misc import derivative
import scipy.integrate as sci

# input the value of n, m, lower and upper limit
#n = int(input("Enter the value of n: "))
#m = int(input("Enter the value of m: "))
n = 2; m = 2;
start = -0.999999; stop = -start
#start = -1; stop = 1
Np = 100000; deriv_spacing = 1e-6; # # spacing=10^-6; Feed it as input from command line
x = np.linspace(start, stop, Np)

# Create Poly1D Chebyshev polynomial and derivatives
tn = chebyt(n); tm = chebyt(m); tnp1 = chebyt(n+1); tnm1 = chebyt(n-1);
tnprime   = derivative(tn,   x, deriv_spacing)
tnp1prime = derivative(tnp1, x, deriv_spacing)
tnm1prime = derivative(tnm1, x, deriv_spacing)

# Logical case switch for different recursion relations to choice from
orthn=1; rect1=1; rect2=1; rect3=1;
#print 'Compare maximum of |lhs-rhs| (L1 norm) to zero for n = ', n

if(orthn): #\int T(n)(x)*T(m)(x)*(1-x**2)^(-1/2) = {0 if m!=n; pi/2 if m=n!=0; pi if
m=n=0}
    I = sci.simps(tn(x)*tm(x)/np.sqrt(1-pow(x,2)),x)
    print 'Orthonormality for n =',n, 'and m =', m, 'is I = ', I, 'which for m=n!=0
is',np.pi/2

if(rect1): #T(n+1)(x) = 2*x*T(n)(x) - T(n-1)(x)
    lhs = tnp1(x)
    rhs = 2.0*x*tn(x) - tnm1(x)
    print ('Maximum of T(n+1)(x)- 2xT(n)(x) + T(n-1)(x) = ', abs(max(lhs-rhs)))

if(rect2): #(1-x^2)*T(n)'(x) = -n*x*T(n)(x) + n*T(n-1)(x)
    lhs = (1-x**2)*tnprime
    rhs = -n*x*tn(x) + n*tnm1(x)
    print ('Maximum of (1-x^2)*Tprime(n)(x) + n*x*T(n)(x) - n*T(n-1)(x) = ', abs(max(lhs-
rhs)))

if(rect3): #2*T(n)(x) = T(n+1)'(x) - 2*x*T(n)'(x) + T(n-1)'(x)
    lhs = 2*tn(x)
    rhs = tnp1prime - 2.0*x*tnprime + tnm1prime
    print ('Maximum of 2*T(n)(x)=dT(n+1)(x)/dx-2xdT(n)(x)/dx + dT(n-1)(x)/dx  = ',
abs(max(lhs-rhs)))
```