```python
# Groundwater Modeling Coding Assignment #2
# Jim Finnegan
# 1D Transport Equation
# Finite Difference Method
import numpy as np
from scipy.sparse import diags
from matplotlib import pyplot as plt

# user inputs
D = float(input('Enter D (m^2/d): '))
R = float(input('Enter R: '))

# other parameters
v, L, dx, t, dt = 0.1, 200, 2, 400, 10
# matrix dimensions
rows = int(t / dt) + 1
cols = int(L / dx) + 1

# initial conditions
C0 = np.zeros(cols)
C = np.zeros((rows, cols))
C[:, 0] = 1 # boundary condition: C/C0 = 1 at x=0

# simplified variables from central difference derivation
G = (D * dt) / (2 * R * dx**2)
H = (v * dt) / (4 * R * dx)
lam_1, lam_2, lam_3, lam_4 = G + H, 2*G + 1, G - H, 2*G - 1

# CENTERED DIFFERENCE SCHEME
#   Left hand side - k+1
A_diagonals = [np.ones(cols-1)*lam_1, np.ones(cols)*-lam_2, np.ones(cols-1)*lam_3]
A = diags(A_diagonals, offsets=[-1, 0, 1], shape=(cols, cols)).toarray()
#   Right hand side - k
B_diagonals = [np.ones(cols-1)*-lam_1, np.ones(cols)*lam_4, np.ones(cols-1)*-lam_3]
B = diags(B_diagonals, offsets=[-1, 0, 1], shape=(cols, cols)).toarray()


for k in range(1, rows):
    b = np.dot(B, C[k-1, :])             # solve RHS
    C[k, :] = np.linalg.solve(A, b)      # solve LHS
    C[k][0] = 1                          # boundary condition
print('Centered difference results: ')
print(C)

x = np.linspace(0, L, num=cols)
plt.plot(x, C[0, :])
plt.plot(x, C[10, :])
plt.plot(x, C[20, :])
plt.plot(x, C[30, :])
plt.plot(x, C[40, :])

```

```
plt.xlabel('distance (m)')
plt.ylabel('C/C0')
plt.legend(['0 days', '100 days', '200 days', '300 days', '400 days'
])
plt.show()

```