```python
# Groundwater Modeling Coding Assignment #2
# Jim Finnegan
# 1D Transport Equation
# Finite Element Method

# SETUP
import numpy as np
from matplotlib import pyplot as plt

# PARAMETERS
# user inputs
D = float(input('Enter D (m^2/d): '))
R = float(input('Enter R: '))
# other parameters
v, L, dx, t, dt = 0.5, 200, 2, 400, 10
# matrix dimensions
rows = int(t / dt) + 1
n_el = int(L / dx)
cols = n_el + 1
# initial conditions
C = np.zeros((rows, cols))
C[:, 0] = 1  # boundary condition: C/C0 = 1 at x=0

# CONSTRUCT STIFFNESS AND STORAGE MATRICES
# element matrices
alpha = (R * dx) / 6
lam_1 = D / dx
lam_2 = v / 2
Ae = [[lam_1 - lam_2, -lam_1 + lam_2], [-lam_1 - lam_2, lam_1 + lam_2]]  # element stiffness matrix
Be = [[2 * alpha, alpha], [alpha, 2 * alpha]]                             # element storage matrix
# global matrices
A = np.zeros((cols, cols))
B = np.zeros((cols, cols))
for i in range(1, cols):
    A[i, i] += Ae[1][1]             # assemble Ae elements
    A[i, i - 1] += Ae[1][0]
    A[i - 1, i] += Ae[0][1]
    A[i - 1, i - 1] += Ae[0][0]
    B[i, i] += Be[1][1]             # assemble Be elements
    B[i, i - 1] += Be[1][0]
    B[i - 1, i] += Be[0][1]
    B[i - 1, i - 1] += Be[0][0]
LH = (A/2 + B/dt)
RH = (-A/2 + B/dt)

# TIME STEPPING
for k in range(1, rows):
    b_f = np.dot(RH, C[k-1, :])             # solve RHS
    b_f[0] = LH[0][0] + LH[0][1]*C[k-1][1]  # boundary condition
    C[k, :] = np.linalg.solve(LH, b_f)      # solve LHS

# PLOT
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, :])
```

```
title_string = 'Finite element solution\n' + 'D = ' + str(D) + ', R = ' + str(R)
plt.title(title_string)
plt.xlabel('distance (m)')
plt.ylabel('C/C0')
plt.legend(['0 days', '100 days', '200 days', '300 days', '400 days'])
plt.show()
```