Build a Matrix Calculator

Description

In this Project, you will use Python to create an application to perform matrix operations. You will need to use the concepts covered in the Python Fundamentals course to complete the project, including:

- Variables and Operators
- Lists
- Loops and Conditional Statements
- Functions and Modules
- Object-Oriented Programming concepts

Step 1: Create an Array Class

Create an `Array` class that has a list member variable to hold the elements of the array. Define the class in a module named array.py

The __init__ method should:

- Take the array size as an argument
- Create a list of the given size with all elements equal to 0

Step 2: Add Array Methods

Given two arrays, defined as follows:

```
a = Array(n1)
b = Array(n2)
```

The user should be able to perform the following operations:

```
a.append(x)      // Append x to the end of the array
a.get(i)         // Get element at index i
a.set(i, x)      // Set element at index i to the value of x
a.contains(x)    // Check if a contains x
a.print()        // Print all elements of a
a.average()      // Calculate the average of the array

a.add(b)
a.subtract(b)
a.multiply(b)    // Element by element multiplication
a.divide(b)      // Element by element division

a.equals(b)      // Check if all elements are equal
```

Step 3: Create a Matrix Class

Create a Matrix class. Define the class in a module named matrix.py

The __init__ method should:

- Take the number of rows and columns of the matrix
- Create a list of arrays with all elements equal to 0

Step 4: Add Matrix Methods

For the following objects:

```
a = Matrix(n1, m1)
b = Matrix(n2, m2)
c = Array(m2)
```

The user should be able to perform the following operations:

```
a.set(r, c, x)   // Set element at r,c to x
a.get(r, c)      // Get element at r,c
a.transpose()    // Transpose the matrix
a.print()        // Print all elements of the matrix

a.add(b)
a.subtract(b)
a.multiply(b)    // Inner dimensions must agree
a.multiply(c)    // Inner dimensions must agree

a.equals(b)      // Check if all elements are equal

a.contains(x)
```

Step 5: Create the Main Module

Create the main module, main.py. In the module:

1. Import the two modules
2. Get the user's requirements
    - Get the number of arrays and or matrices to be created
    - Read for the dimensions of the arrays and matrices to be created
3. Create objects of the two classes according to the user's requirements
4. Ask the user for the number of operations to be performed. For each:
    - Ask for the operands of the operation
    - Give the user a list of valid operations
    - Read the required operation
5. Complete all calculations and show step by step results to the user