Programming Assignment #1

MAT 399A, Fall 2025

Due Friday, September 12

Python installation

Install Python 3 on your system if it is not already installed. On Windows, you can either download the Windows installer from

```
www.python.org
```

or use WSL. If you are using Linux or WSL, you can type

```
sudo apt update
sudo apt install python3
sudo apt install python3-pip
```

from a terminal. Note that if you are running Linux, it is recommended that you run Python in a virtual environment. You can do this by navigating to your working folder and typing

```
python3 -m venv .venv
source .venv/bin/activate
```

We will be using the following Python packages, so you will need to install them on your system if they are not installed already:

`numpy` — a standard numerical linear algebra package

`qiskit` — IBM's quantum circuit package

`qiskit_aer` — IBM's quantum computation emulator

This can be done using the Python package installer `pip`. E.g., for the `numpy` package you can issue

```
pip install numpy
```

from the command line. The remaining two packages can be installed similarly.

Once you have installed the requisite packages, you can use the script `test_all.py` to test your installation. If you run

```
python test_all.py
```

you should get the same output as in the file `test_all.output.txt`. Note that the "counts" in the last line of output that you get will probably be somewhat different (but not too different) from the values in the text file.

Programming tasks

1. Write a function to compute the Hermitian conjugate (also called adjoint) of a square matrix.

```
def adjoint(M):
  """
  where:
    M : numpy square matrix
  returns:
    Hermitian conjugate of M
  """
```

2. Write a function to compute the Hermitian inner product of two square matrices.

```
def inner(A,B):
  """
  where:
    A,B : numpy square matrices
  returns:
    Hermitian inner product of A and B
  """
```

   Programming challenge: write the function so that works both with square matrices and with arrays of the same dimension.

3. Write a function to test if a square matrix is unitary.

```
def isUnitary(A):
  """
  where:
    A: numpy square matrix
  returns:
    True if A is unitary, False otherwise
  """
```

   Note that because of finite numerical precision, you should not compare a floating point number to an exact value such as 0. For the purposes of this assignment, you can assume that a floating point number less than `1e-20` is close enough to 0.

What to turn in

The code for all of the above functions should be put in a single file named `exercise1.py`. You may only import the packages `numpy`, `math`, and `cmath`.

Test driver

A simple test driver `exercise1_test.py` has been uploaded for convenience in testing. If you run

```
python exercise_test.py
```

the output should match the contents of the file `exercise1_test.output.txt` to within reasonable numerical precision.

The test driver is intentionally minimal. It is recommended that you modify the driver to make more extensive tests of your code.