# Lab 2: Matrices and 1-Qubit Gates

## Preliminaries

Refer to teaching materials in Module 2.

## Tasks

Write your answers for all tasks to your lab report.

1. Start 'jupyter notebook', and create a notebook for this lab.

   NB: This task doesn't need an answer in the lab report.

2. Use three matrices A, B, and C as an example, show that matrix multiplication satisfies distributivity on the right: (A+B)C = AC + BC. This should be verified by both manual calculation and Python code. The three matrices you pick should be different from the ones used in the Jupyter NB accompanying our lecture.
   2.1 Use Markdown cells and Latex to include the manual calculation for the above.
   2.2 Use Markdown cells and Code cells to include the Python code for the above.

3. Use three matrices A, B, and C as an example, show that matrix multiplication satisfies associativity: (AB)C = A(BC). This should be verified by both manual calculation and Python code. The three matrices you pick should be different from the ones used in the Jupyter NB accompanying our lecture.
   3.1 Use Markdown cells and Latex to include the manual calculation for the above.
   3.2 Use Markdown cells and Code cells to include the Python code for the above.

4. Use two matrices A and B as an example, show that matrix multiplication may not satisfy commutativity: AB = BA. This should be shown by both manual calculation and Python code. The three matrices you pick should be different from the ones used in the Jupyter NB accompanying our lecture.
   4.1 Use Markdown cells and Latex to include the manual calculation for the above.
   4.2 Use Markdown cells and Code cells to include the Python code for the above.

5. Use Python code to find the inverse of the matrix $\begin{bmatrix} 2 & 3 \\ 4 & 5 \end{bmatrix}$ and print the result.

6. Create a QuantumCircuit with a 1-qubit input and no bit output. Initialize its qubit $q_0$ with the state of $\frac{1}{3}|0\rangle + \frac{2\sqrt{2}}{3}|1\rangle$. Then, apply X-gate to $q_0$.

   6.1 Draw the above circuit.
   6.2 Use a StatevectorSimulator to run the above circuit with one shot only. After the run, obtain the state vector of $q_0$ and print it.

7. Create a QuantumCircuit with a 1-qubit input and a 1-bit output. Initialize its qubit $q_0$ with the state of $-\frac{1}{3}|0\rangle + \frac{2\sqrt{2}}{3}|1\rangle$. Then, apply X-gate, Z-gate, and H-gate to $q_0$ in turn (NB: follow this order exactly). Finally, measure $q_0$.
   7.1 Draw the above circuit.
   7.2 Use a QasmSimulator to run the above circuit with 2000 shots, and then plot the results with a histogram.
   7.3 According to your histogram, what is the approximate probabilities of obtaining 0 and 1 respectively?
   7.4 Use a Markdown cell to include manual calculation on the state of $q_0$ after those three gates are applied.
   7.5 Verify that the approximate probability distribution obtained in 7.3 conforms with the state of $q_0$ obtained in 7.4. This should be done in a Markdown cell with the calculations based on probability amplitudes.