Assignment 1

Problem Statement:

Implementations of 16 Qubit Random Number Generator.

Objective:

1. Understand creation of Qubit circuit
2. Create 16 Qubit Random Number Generator

Outcome:

Displays 16 Qubit Random Number

Theory:

To create a Random Number Generator in qiskit for IBMs quantum computers using 16 qubits.

Requirements:

Python 3.x or above (available here: https://www.python.org/)

Pip: A package management system for Python (included with Python 3.x)

IBM Q Account: This is so you can run your programs on IBM quantum devices. You can sign up for one here: https://quantum-computing.ibm.com

Installation :

Install Python 3.x (Make sure Python is added to Path and Pip is checked)

Open Command Prompt and type in: pip install qiskit

Steps to perform:

STEP 1 : INITIALISE THE QUANTUM AND CLASSICAL REGISTERS

The first step is to initialise a 16 qubit register . This is done by the following code:

```
q = QuantumRegister(16,’q’)
```

Next we initialise the 16 bit classical register with the following code:

```
c = ClassicalRegister(16,’c’)
```

STEP 2 : CREATE THE CIRCUIT

Next we create a quantum circuit using the following code:

```
circuit = QuantumCircuit(q,c)
```

STEP 3 : APPLY A HADAMARD GATE TO ALL QUBITS

Then we need to apply a Hadamard gate. This gate is used to put a qubit in to a superposition of 1 and 0 such that when we measure the qubit it will be 1 or a 0 with equal probability.

This is done with the following code:

```
circuit.h(q)
```

STEP 4 : MEASURE THE QUBITS

After this we measure the qubits. This measurement will collapse the qubits superposition in to either a 1 or a 0.

This is done with the following code:

```
circuit.measure(q,c)
```

ALGORITHM :

1. Start
2. pip install qiskit
3. Initialise the quantum and classical registers
4. Create the circuit
5. Apply a hadamard gate to all qubits
6. Measure the qubits
7. Stop

How to run the program :

Write the code in to a python file.
Enter your API token in the IBMQ.enable_account('Insert API token here') part
Save and run.

Conclusion:

By this way, we can generate 16 Qubit Random Number.

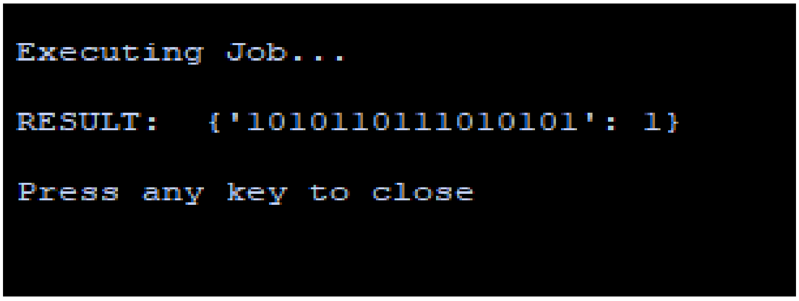
Oral Questions:

1. What is Qubit?
2. What are different steps performed to generate random number?
3. What are different types of registers?
4. What is quantum circuit?
5. How you measure the qubits?
6. What is Hadamard gate?

Code :

```
from qiskit import QuantumRegister, ClassicalRegister,
QuantumCircuit, execute,IBMQ

IBMQ.enable_account('ENTER API TOKEN HERE')
provider = IBMQ.get_provider(hub='ibm-q')

q = QuantumRegister(16,'q')
c = ClassicalRegister(16,'c')
circuit = QuantumCircuit(q,c)
circuit.h(q) # Applies hadamard gate to all qubits
circuit.measure(q,c) # Measures all qubits

backend = provider.get_backend('ibmq_qasm_simulator')
job = execute(circuit, backend, shots=1)

print('Executing Job...\n')
result = job.result()
counts = result.get_counts(circuit)

print('RESULT: ',counts,'\n')
print('Press any key to close')
input()
```

Output :



```
Executing Job...

RESULT:  {'1010110111010101': 1}

Press any key to close
```

Implementation :

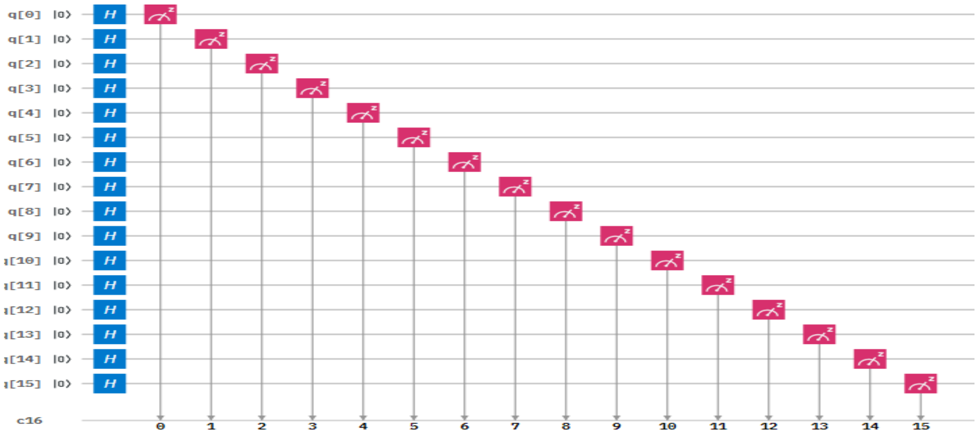


Figure 1: Circuit Diagram of the 16-qubit Random Number Generator