**Programming Quantum bits with IBM Tool “Qiskit “**

IBM is making quantum computing more accessible than ever before with open source developer tool called “qiskit” to experience the quantum programming from anywhere of the world. Thousands of programs are already running from all over the world, including Antarctica.

Here is how it works; information can be entered through the classical computer which travels through the quantum computer via internet and reaches to the quantum computer hosted on the IBM cloud. The information can converts into frequencies of microwave pulses that can change the states of Q-bit. This change can be stored in the form of information. As the information travels through the room temperature, it will further changes into microwave pulses and their temperature gets down to -273 degree Celsius to interact with q-bits. This process takes only 0.0001 second. After the successful interaction with q-bits, the final result sent back through the internet in the form of classical computer data.

The quantum computer is capable to solve the new class of science and business problems.

To try Quantum Computing and work with “qiskit” tool, follow the instruction given below.

**Installation and execute**

**Requeriments**

* Have installed [Python](https://www.python.org/downloads/) 3.5 or above and [Jupyter notebooks](http://jupyter.org) or [Jupyter lab](https://github.com/jupyterlab/jupyterlab), we recomend use Anaconda because provide to you all the elements.
* Install Qiskit: pip install qiskit

**1. Dependencies**

To use QISKit you’ll need to have installed at least Python 3.5 or later. Jupyter Notebooks is also recommended for interacting with tutorials.

For this reason we recommend installing Anaconda 3 python distribution, which already comes with all these dependencies pre-installed.

**2. Installation**

The recommended way to install QISKit is by using the PIP tool (Python package manager):

pip install qiskit

This will install the latest stable release along with all the dependencies.

**3. Configure your API token and QE credentials**

Create an IBM Q experience account if you haven’t already done so

Get an API token from the IBM Q experience website under “My Account” > “Personal Access Token”

The API token needs to be placed in a file called Qconfig.py. For convenience, we provide a default version of this file that you can use as a reference: Qconfig.py.default. After downloading that file, copy it into the folder where you will be invoking the SDK (on Windows, replace cp with copy):

cp Qconfig.py.default Qconfig.py

Open your Qconfig.py, remove the # from the beginning of the API token line, and copy/paste your API token into the space between the quotation marks on that line. Save and close the file.

For example, a valid and fully configured Qconfig.py file would look like:

APItoken = '123456789abc...'

config = {

'url': 'https://quantumexperience.ng.bluemix.net/api'

}

If you have access to the IBM Q features, you also need to setup the values for your hub, group, and project. You can do so by filling the config variable with the values you can find on your IBM Q account page.

For example, a valid and fully configured Qconfig.py file for IBM Q users would look like:

APItoken = '123456789abc...'

config = {

'url': 'https://quantumexperience.ng.bluemix.net/api',

# The following should only be needed for IBM Q users.

'hub': 'MY\_HUB',

'group': 'MY\_GROUP',

'project': 'MY\_PROJECT'

}

**Install Jupyter-based tutorials**

The QISKit project provides you a collection of tutorials in the form of Jupyter notebooks, which are essentially web pages that contain “cells” of embedded Python code. Please refer to the tutorials repository for detailed instructions.

**Troubleshooting**

The installation steps described on this document assume familiarity with the Python environment on your setup (for example, standard Python, virtualenv or Anaconda). Please consult the relevant documentation for instructions tailored to **your** environment.

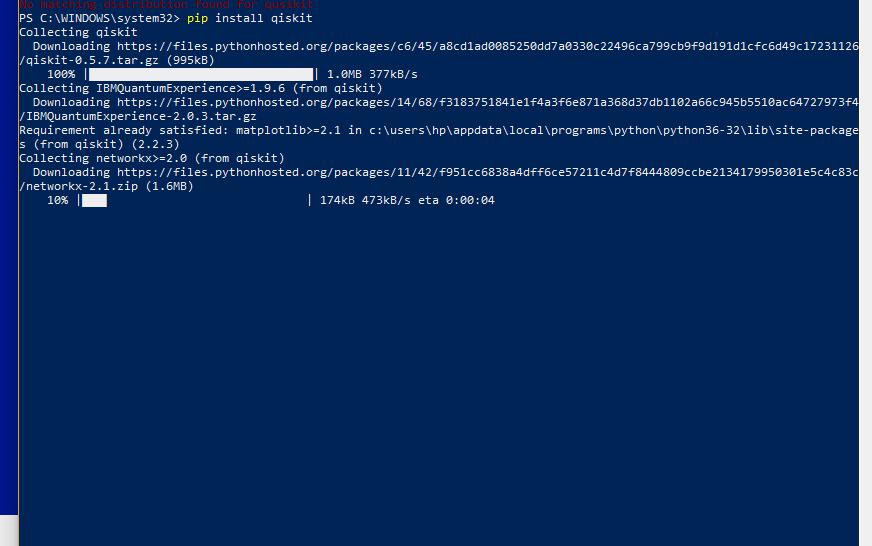
Depending on the system and setup, appending “sudo -H” before the pip install command could be needed:

pip install -U --no-cache-dir qiskit

For additional troubleshooting tips, see the QISKit troubleshooting page on the project’s GitHub wiki.

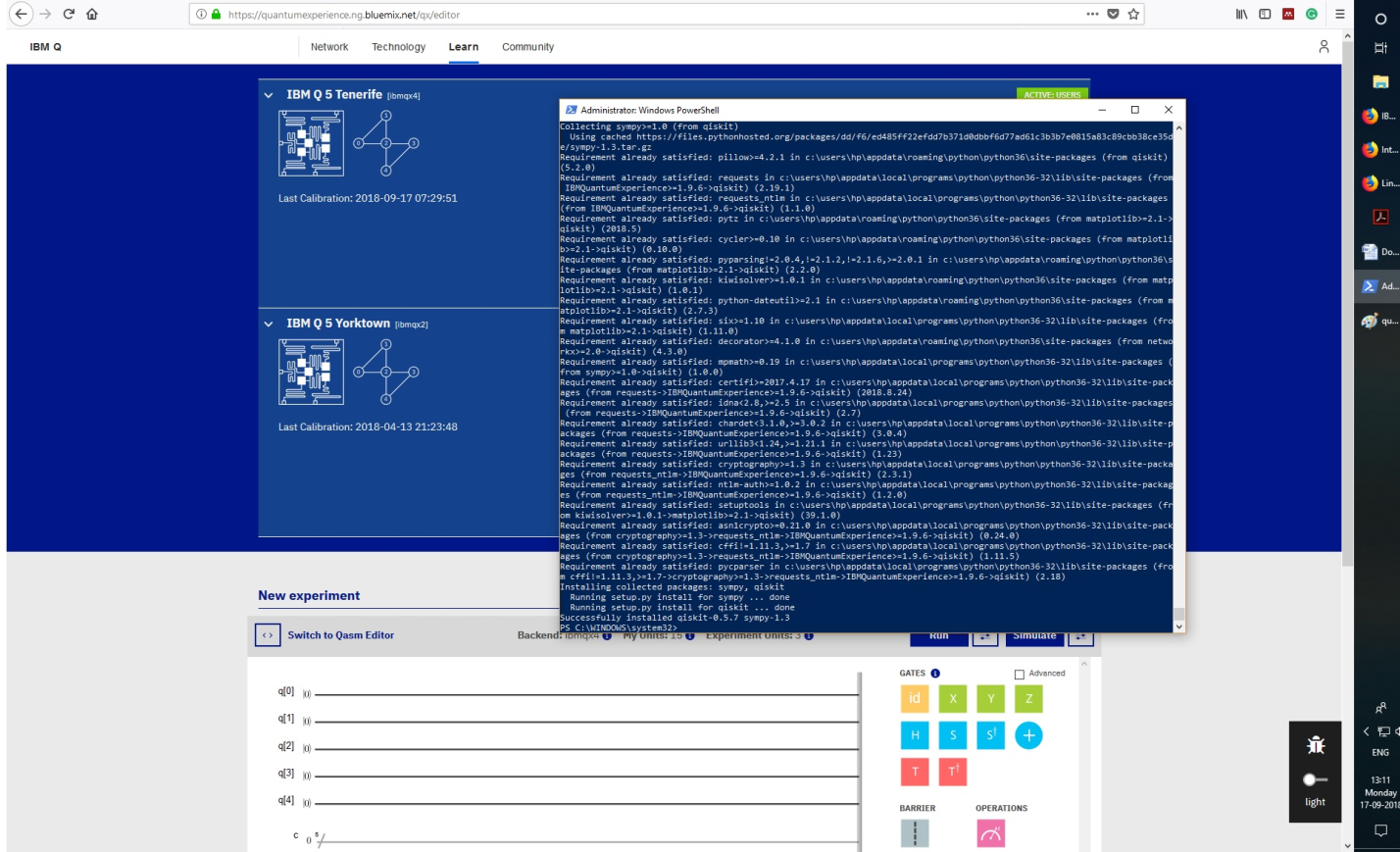
Example : First Program...

1. qiskit installation..



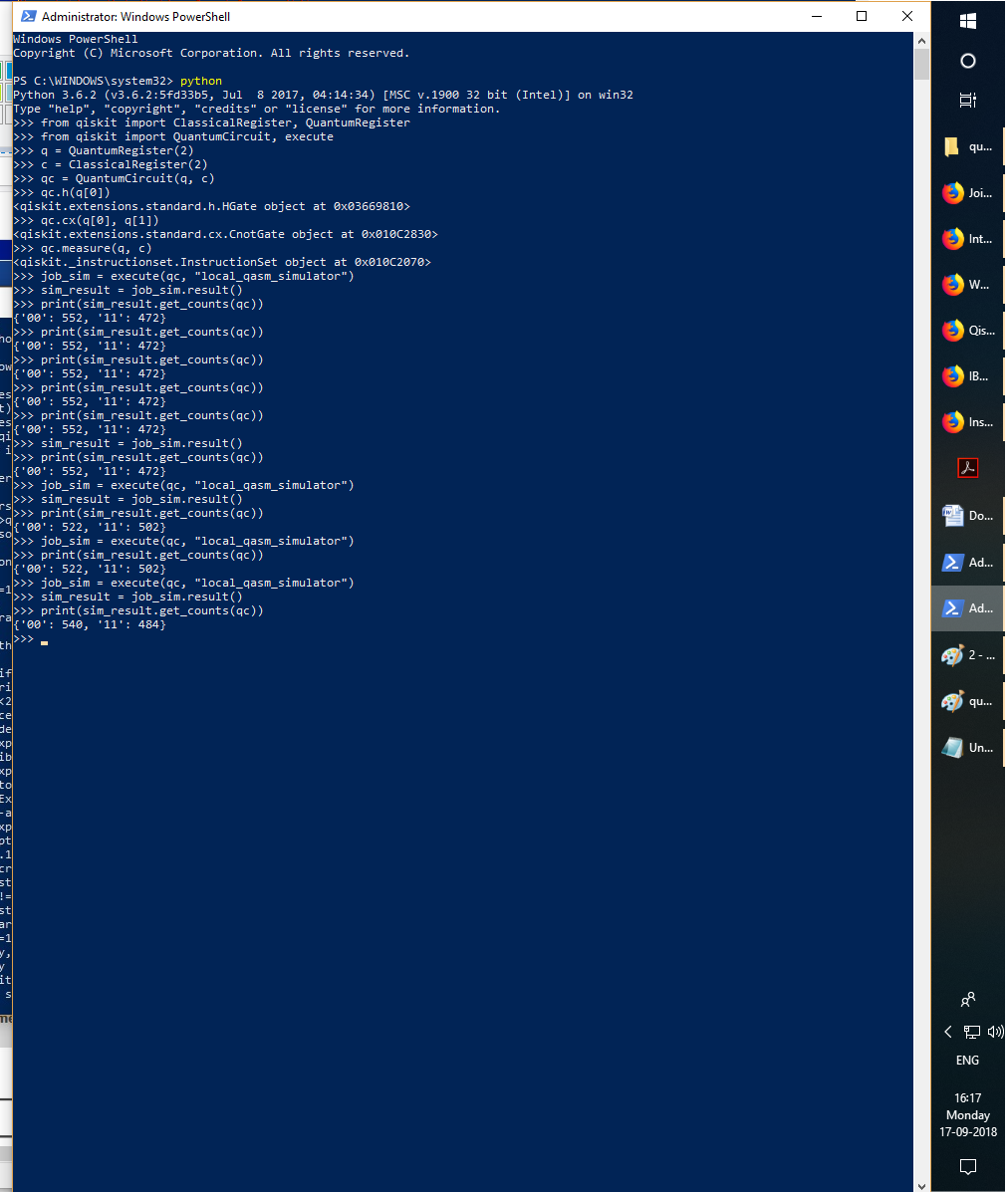
It will take some time.....

Installation completed



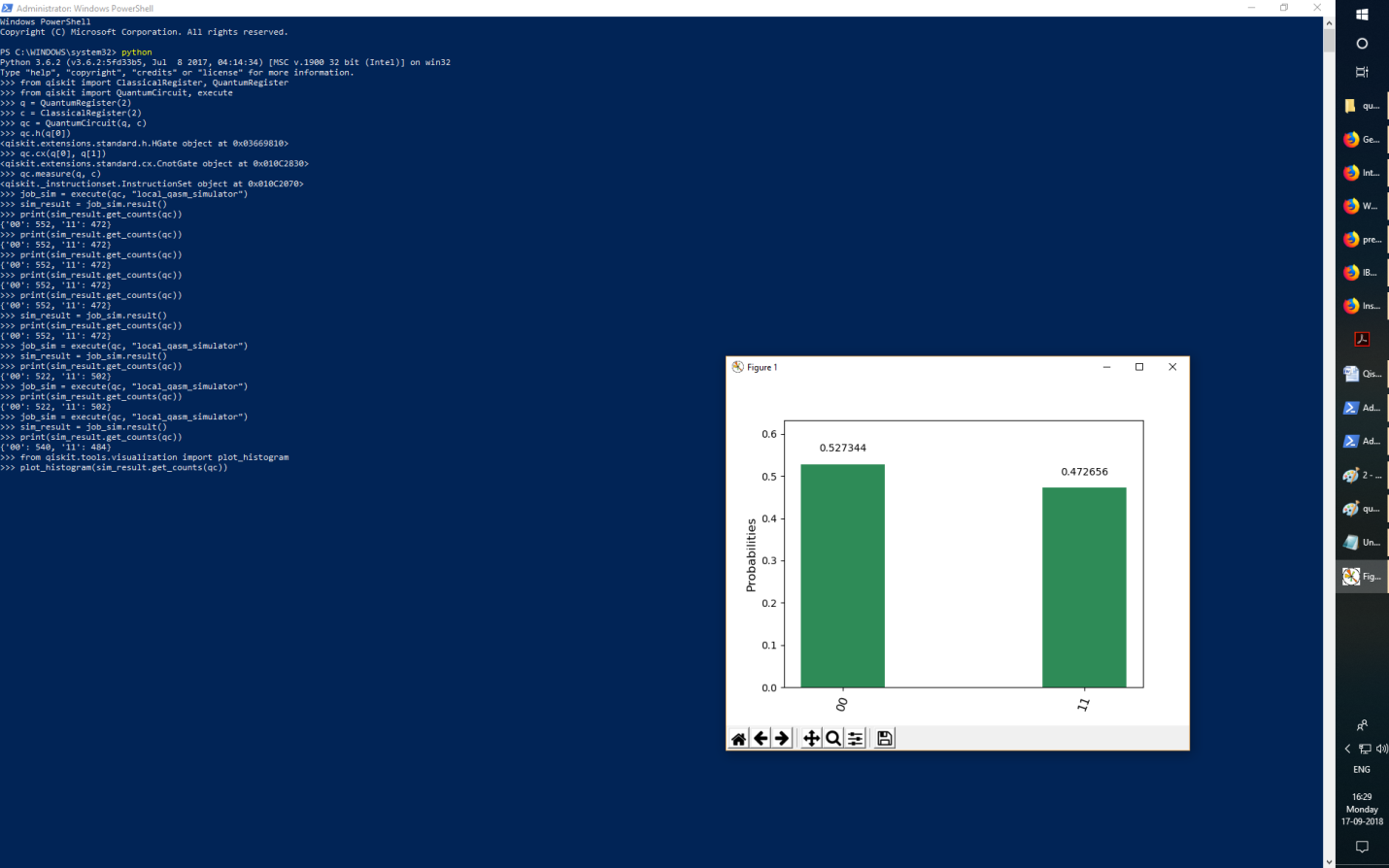
Now start programming...

Test Program:



qiskit is working properly...!!!!

Plotting the results:



Its done...!!!!!