

Hackathon Prompt: Real-world Applications

Background and motivation:

We've entered a new era of quantum computing - The Quantum Utility era.

Quantum utility is what we get when a quantum computer can perform reliable computations at a scale beyond brute force classical computing methods that provide exact solutions to computational problems. Now, computational scientists and other researchers can tackle these large-scale problems using quantum computers with IBM's 100+ qubits backends accessible to everybody. Entering the era of quantum utility is, in other words, the quantum computers we have today are valuable tools researchers can use to explore meaningful scientific problems. Now it is your turn to tackle and explore issues with this leading-edge computational resource.

This challenge aims to design and build a cloud-based, quantum-powered application that addresses a real-world problem and is accessible to end users. This includes applications of quantum algorithms that can have practical usage and, in theory, be exposed to businesses or individual users on the internet for consumption. Teams should identify a problem that can be solved (though not necessarily more efficiently) with quantum computers.

Your prompt is to create a quantum-powered application that utilizes at least 80 qubits.

Deeper questions:

We encourage contestants to be creative and to leverage their existing quantum knowledge to develop new applications and experiments. To get you started, we offer some suggestions and prompts that may lead to interesting projects.

Some examples include but are not limited to:

- An application involving a Random Number Generator
- An application involving Optimization
- An application involving Chemical Simulation
- An application involving a Quantum Calculator
- An application involving Image Classification
- A game involving a quantum algorithm

Suggested resources:

- [Variational algorithm design](#)
- [Solve utility-scale quantum optimization problems](#)
- [Quantum Computing in Practice playlist](#)