Quantum Snacks

Challenge information

Computers speak the language of binary (zeros and ones), and every computer program is basically just applying logic gates (e.g NOT, OR, AND, XOR, etc) to these bits to get the desired outcome. The important point here is to remember that a bit is a single variable that can have one of 2 values, either zero or one).

Quantum computers speak a very different language. Instead of a bit, we have a qubit (quantum bit), which can be represented as a two-dimensional vector. Classical bit 0 is equivalent to the qubit, and the classical bit 1 is equivalent to the qubit, but there are many other qubit states that do not have a classical equivalence, which means that qubits can store much more information than classical bits. The goal of this challenge is to understand this concept better.

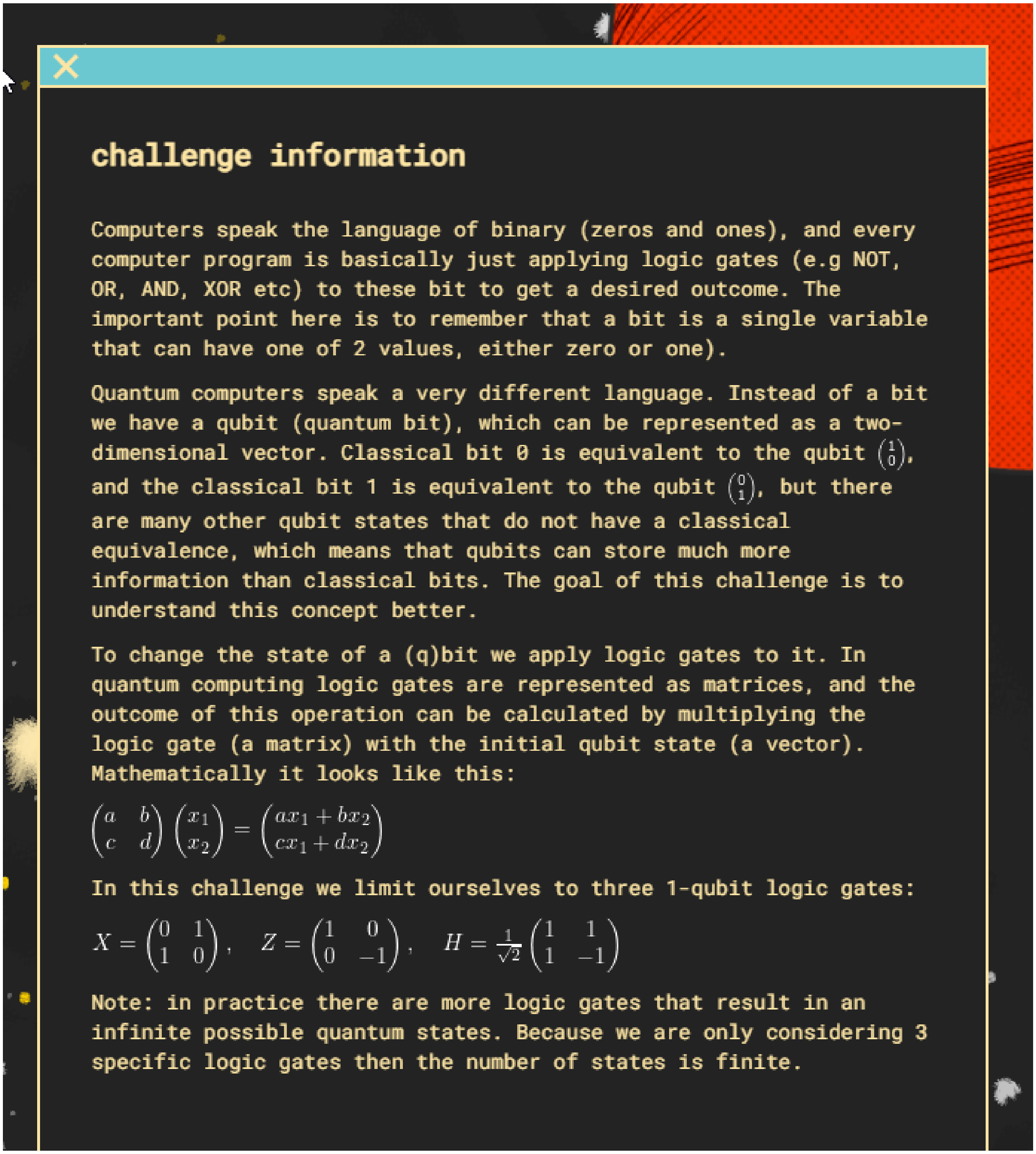
To change the state of a (q)bit we apply logic gates to it. In quantum computing, logic gates are represented as matrices, and the outcome of this operation can be calculated by multiplying the logic gate (a matrix) with the initial qubit state (a vector). Mathematically it looks like this:

$$\begin{pmatrix} a & b \\ c & d \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} = \begin{pmatrix} ax_1 + bx_2 \\ cx_1 + dx_2 \end{pmatrix}$$

In this challenge we limit ourselves to three 1-qubit logic gates:

$$X = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}, \quad Z = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}, \quad H = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}$$

Note: in practice, there are more logic gates that result in infinite possible quantum states. Because we are only considering 3 specific logic gates then the number of states is finite.



challenge information

Computers speak the language of binary (zeros and ones), and every computer program is basically just applying logic gates (e.g NOT, OR, AND, XOR etc) to these bit to get a desired outcome. The important point here is to remember that a bit is a single variable that can have one of 2 values, either zero or one).

Quantum computers speak a very different language. Instead of a bit we have a qubit (quantum bit), which can be represented as a two-dimensional vector. Classical bit 0 is equivalent to the qubit $\binom{1}{0}$, and the classical bit 1 is equivalent to the qubit $\binom{0}{1}$, but there are many other qubit states that do not have a classical equivalence, which means that qubits can store much more information than classical bits. The goal of this challenge is to understand this concept better.

To change the state of a (q)bit we apply logic gates to it. In quantum computing logic gates are represented as matrices, and the outcome of this operation can be calculated by multiplying the logic gate (a matrix) with the initial qubit state (a vector). Mathematically it looks like this:

$$\begin{pmatrix} a & b \\ c & d \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} = \begin{pmatrix} ax_1 + bx_2 \\ cx_1 + dx_2 \end{pmatrix}$$

In this challenge we limit ourselves to three 1-qubit logic gates:

$$X = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}, \quad Z = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}, \quad H = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}$$

Note: in practice there are more logic gates that result in an infinite possible quantum states. Because we are only considering 3 specific logic gates then the number of states is finite.

1> How many states?

Considering only these 3 logic gates, and starting in the

state, how many states can the qubit have? Hint: Start applying the logic gates in random order. Drawing the result in x-y coordinates will greatly help you understand the result.

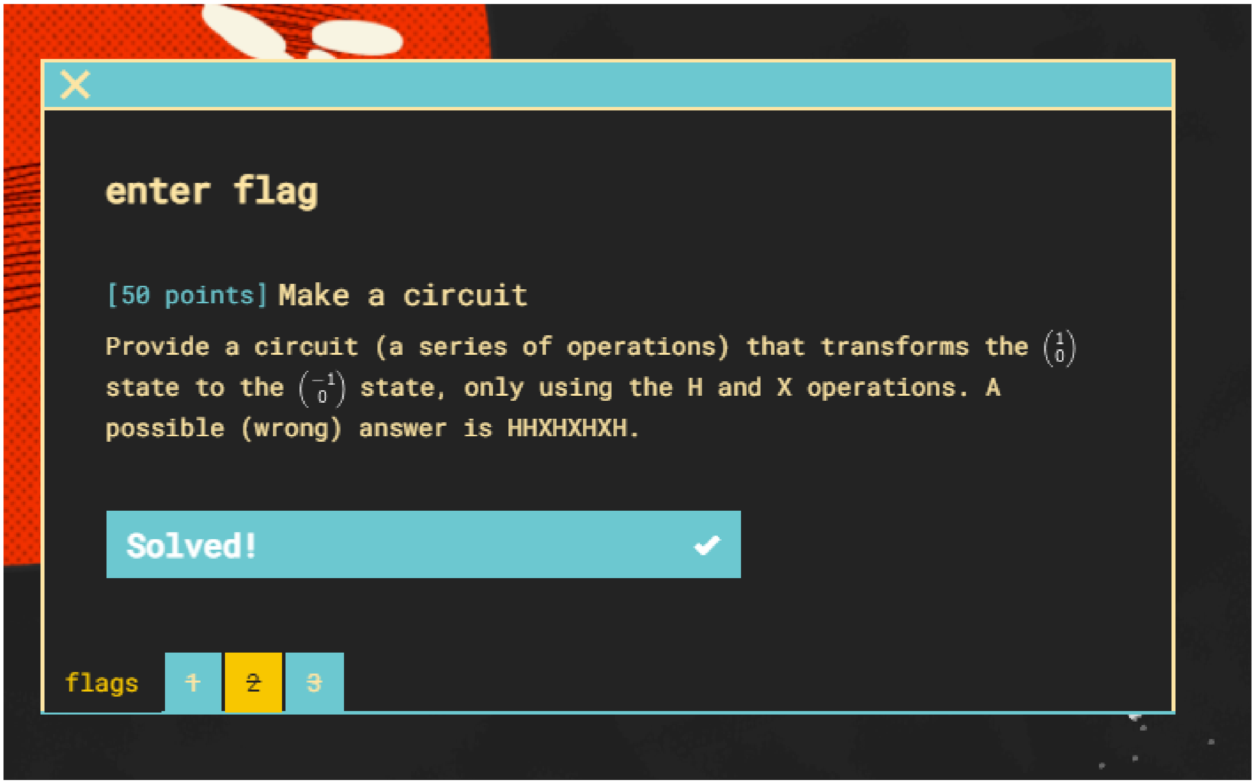
Points [50 points]

- In this challenge, we have to find out the no of states that a logic gate can have since each qubit can be in 2 states at a time so if there are 3 logic gates then the total no of gates is equal to 2^3 = 8
- This is the First Flag
- This behavior is due to the superposition of qubits and can represent all 8 states at the same time.

Flag:- 8

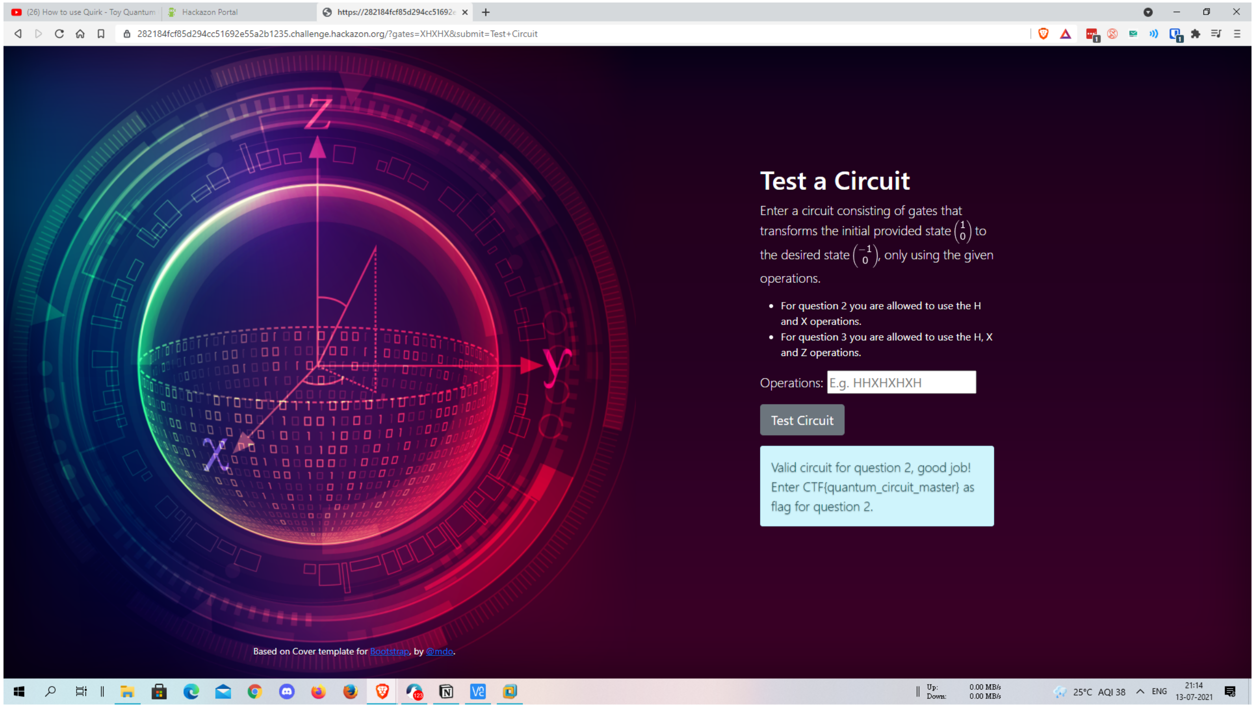
2> Make a circuit

Provide a circuit (a series of operations) that transforms the state to the state, only using the H and X operations. A possible (wrong) answer is HHXHXHXH.



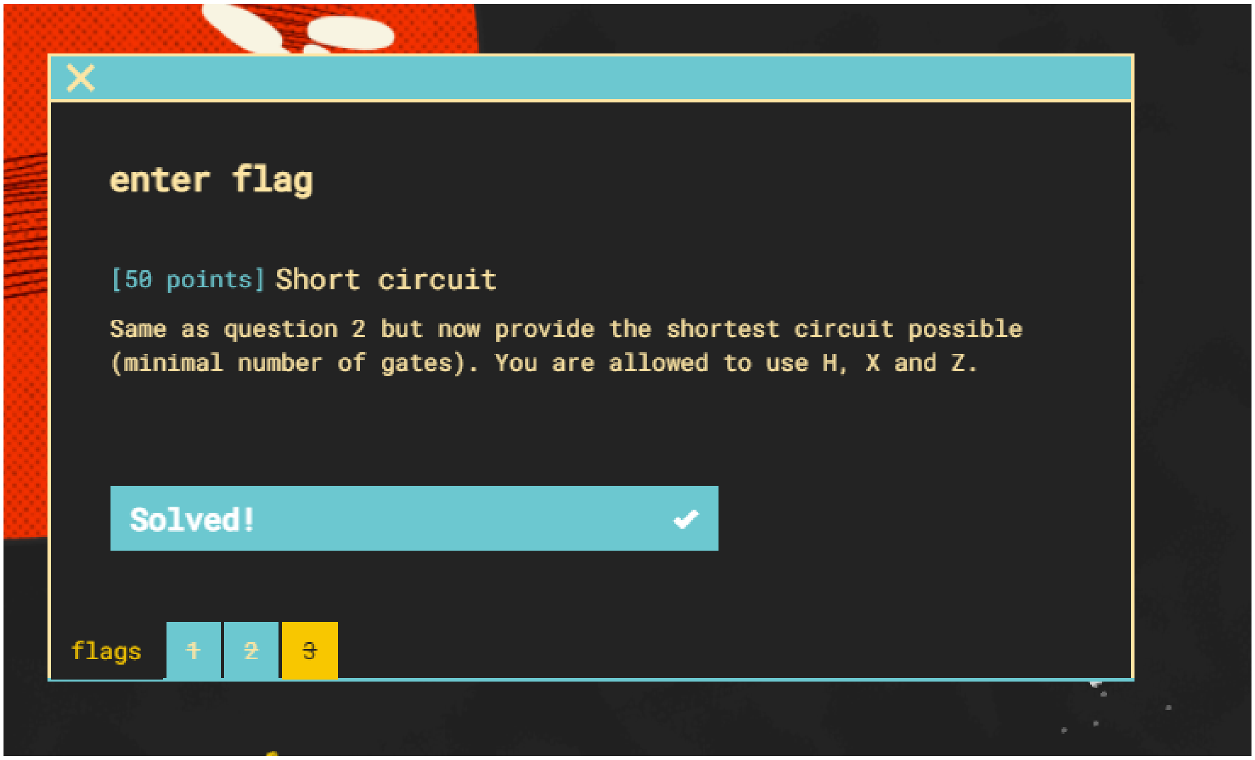
Points [50 points]

- In this, we have to use the above matrix values and the qubit value assume them as matrix multiplication and perform the operations, so I did perform the operation and found out a perfect combination of these operations which will give me the solution, In this challenge, we have to use only H and X operation to get the result.
- Series of operations:- XHXHX
- When we enter this operation in the website provided we get the flag for the second question.
- Flag :- CTF{quantum_circuit_master}



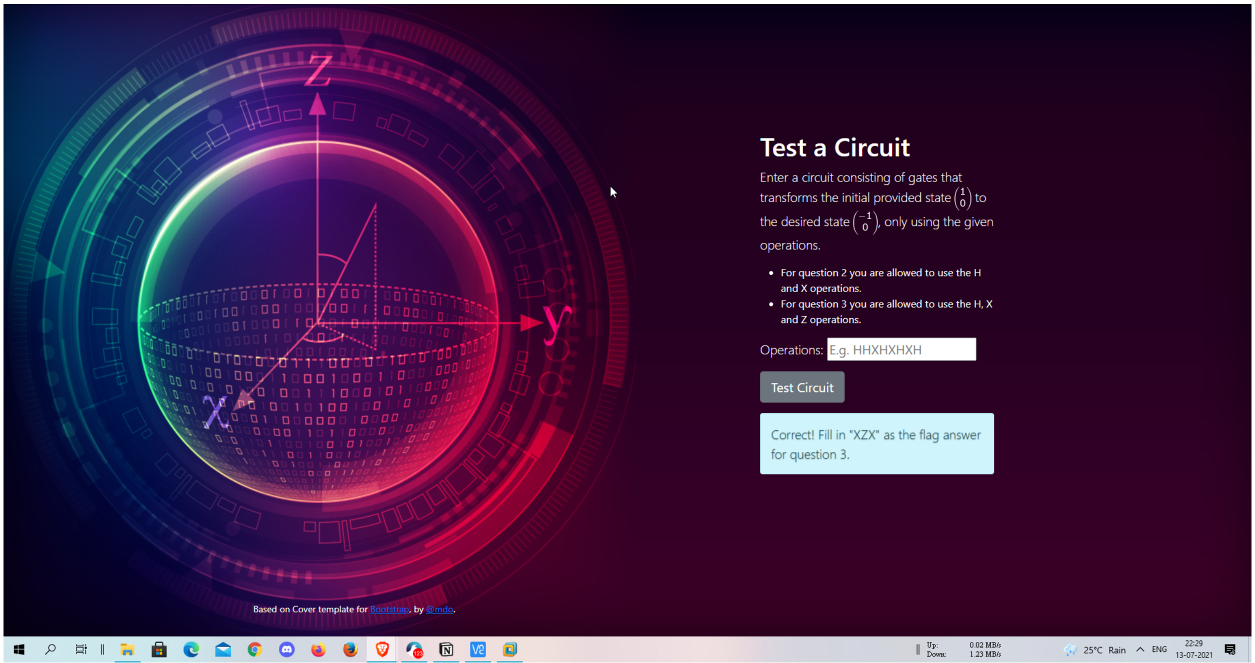
3> Short circuit

Same as question 2 but now provide the shortest circuit possible (minimal number of gates). You are allowed to use H, X, and Z.



Points [50 points]

- This challenge is similar to the 2nd challenge but allows us to use Z operations also to get the solution we also have to find out the smallest solution
- The Series of combinations is:- XZX



Z
y
x
Test a Circuit
Enter a circuit consisting of gates that transforms the initial provided state $\binom{1}{0}$ to the desired state $\binom{-1}{0}$, only using the given operations.
For question 2 you are allowed to use the H and X operations.
For question 3 you are allowed to use the H, X and Z operations.
Operations:
E.g. HHXHXHXH
Test Circuit
Correct! Fill in "XZX" as the flag answer for question 3.
Based on Cover template for Bootstrap, by @mdo.

- Flag :- XZX