😄

Quantum Shuttle

challenge information

In Quαntum Snαcks we discussed qubits αnd single qubit gαtes. In this exercise, we will move on to multiple qubits.

For this question, we will look at the controlled NOT-gate. In the classical controlled-NOT, two bits are input and the output can be described as follows: the first bit (control bit) does not change, the second bit (target bit) flips (NOT-gate) if the control bit is 1.

Example

`CNOT(10)`, the control bit is 1, the target bit is 0. So the output would be: `CNOT(10) = 11`, since the control bit does not change but the target bit flips (since the control bit is 1).

We can also consider the controlled-not operator where the first bit is the target bit and the second bit is the control bit. We denote this with `NOTC`. So for example `NOTC(11) = 01`.

challenge information

In Quantum Snacks we discussed qubits and single qubit gates. In this exercise we will move on to multible qubits.

For this question we will look at a the controlled NOT-gate. In the classical controlled-NOT two bits are input and the output can be described as follows: the first bit (control bit) does not change, the second bit (target bit) flips (NOT-gate) if the control bit is 1.

Example

`CNOT(10)`, the control bit is 1, the target bit is 0. So the output would be: `CNOT(10) = 11`, since the control bit does not change but the target bit flips (since the control bit is 1).

We can also consider the controlled-not operator where the first bit is the target bit and the second bit is the control bit. We denote this with `NOTC`. So for example `NOTC(11) = 01`.

BACKEND SYSTEMS (SHARED)

Backend systems are running for you.
This environment will run until 2022-07-15 23:06:10 CET
Note: the infrastructure of this challenge is shared between multiple players, please be careful.

Use the following link to access the challenge:
https://0ea4d44af418f983943bf03c3f10f998.challenge.hackazon.org/
Note: this is a challenge with intentionally vulnerable systems and applications, in which you are allowed to find vulnerabilities. For more information click here.

1> Bitstring

Consider the following bit-string `b = 000101001011`, and denote *b[i]* to be the *i*-th digit of *b*

What is the resulting bit-string generated by the following protocol

`CNOT(b[1]b[2])NOTC(b[3]b[4])CNOT(b[7]b[5])NOTC(b[6]b[8])CNOT(b[11]b[12])NOTC(b[10]b[9])`

Note: you do not need to use the verification tool for this subtask. You can enter the bitstring directly as the flag.

Update 16/07: fixed flag submission.

Points [50 points]

Solution :-

- The challenge is a continuation to the qbit calculations that we have done in the previous phase 1 in this we have to use the 2 bit gates which are CNOT and NOTC , to get a little more information on how the calculation will work I referred the video given below .

- Reference Links :- https://www.youtube.com/watch?v=rLF-oHaXLtE&t=239s
- So in CNOT gate the the control bit determines if the Target bit is to be flipped (just as we do for a simple not gate) if the control bit is set i.e. 1 then we have to flip the target bit . In CNOT gate the first bit is the control bit while the second bit is the target bit .
- NOTC gate is similar to the CNOT gate only difference is that the First bit is the Target bit and Second bit the control bit . while operation remains same .
- Now in the challenge we are told to certain set of operations on the bit string and concatenate the resulting string which will be the flag for the challenge .
- The Calculation is given below
- CNOT(b[1]b[2]) = CNOT(00) = 00
- NOTC(b[3]b[4]) = NOTC(01) = 11
- CNOT(b[7]b[5]) = CNOT(00) = 00
- NOTC(b[6]b[8]) = NOTC(10) = 10
- CNOT(b[11]b[12]) = CNOT(11) = 10
- NOTC(b[10]b[9]) = NOTC(01) = 11
- Flag :- 001100101011

2> CNOT

Before you continue, have a look at the background in the PDF attached to this challenge.

In the classical situation the `CNOT` gate always keeps the control bit the same but changes the target bit.

Refer to the previous Quantum Snacks exercise (subtask 1) where we had a number different qubit states. For this exercise, we restrict ourselves to the states that you identified for that subtask.

Can you come up with a combination (pair) of qubit states which leave the target bit the same but changed the control bit?

Write your answer (using the verification tool) as a list of numbers that correspond to the below equation: [A,B,C,D,E,F]

$$\left(\frac{1}{\sqrt{A}}\begin{pmatrix}B\\C\end{pmatrix}\right)\otimes\left(\frac{1}{\sqrt{D}}\begin{pmatrix}E\\F\end{pmatrix}\right)$$

Points [50 points]

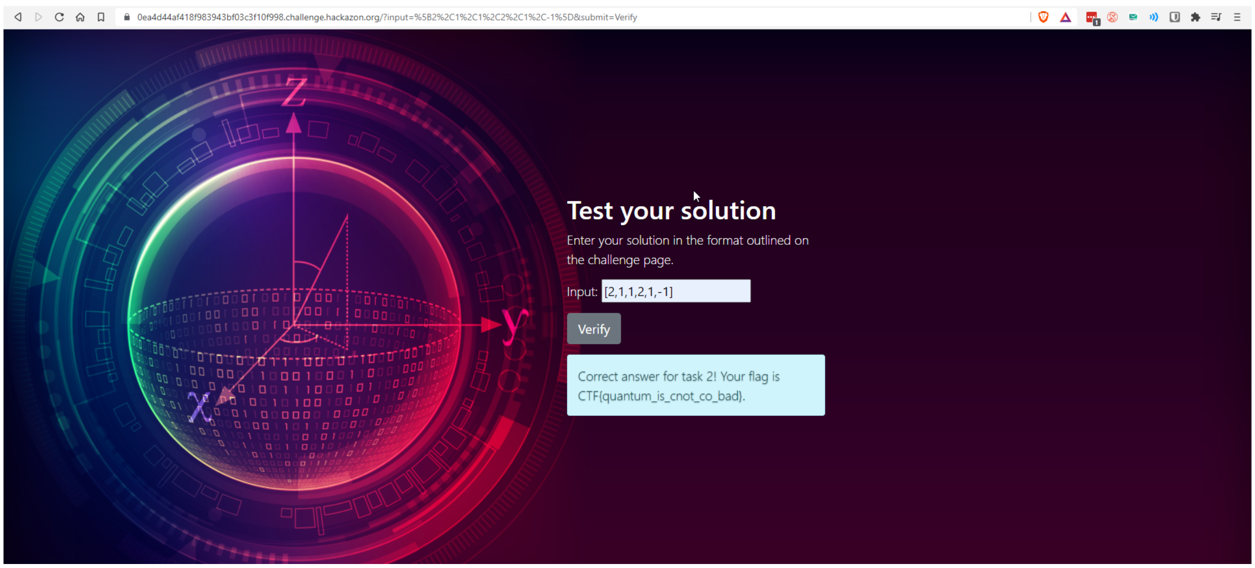
Solution

- In this challenge we have to find the values of A B C D E F in such a way that their tensors product will be that the control bit is change while the target bit remains same to do this I have identified that the H matrix from Quantum snacks challenge when used with itself to calculate tensor product will give us the resulting operation therefore the values of [2,1,1,2,1,-1]

  Reference Link:- https://docs.microsoft.com/en-us/azure/quantum/concepts-multiple-qubits

- It is the operation H * H which will give us the desired value

- When the above value is entered in the validator website we get the flag



Flag :- CTF{quantum_is_cnot_co_bad}

Reference Link:- https://docs.microsoft.com/en-us/azure/quantum/concepts-multiple-qubits

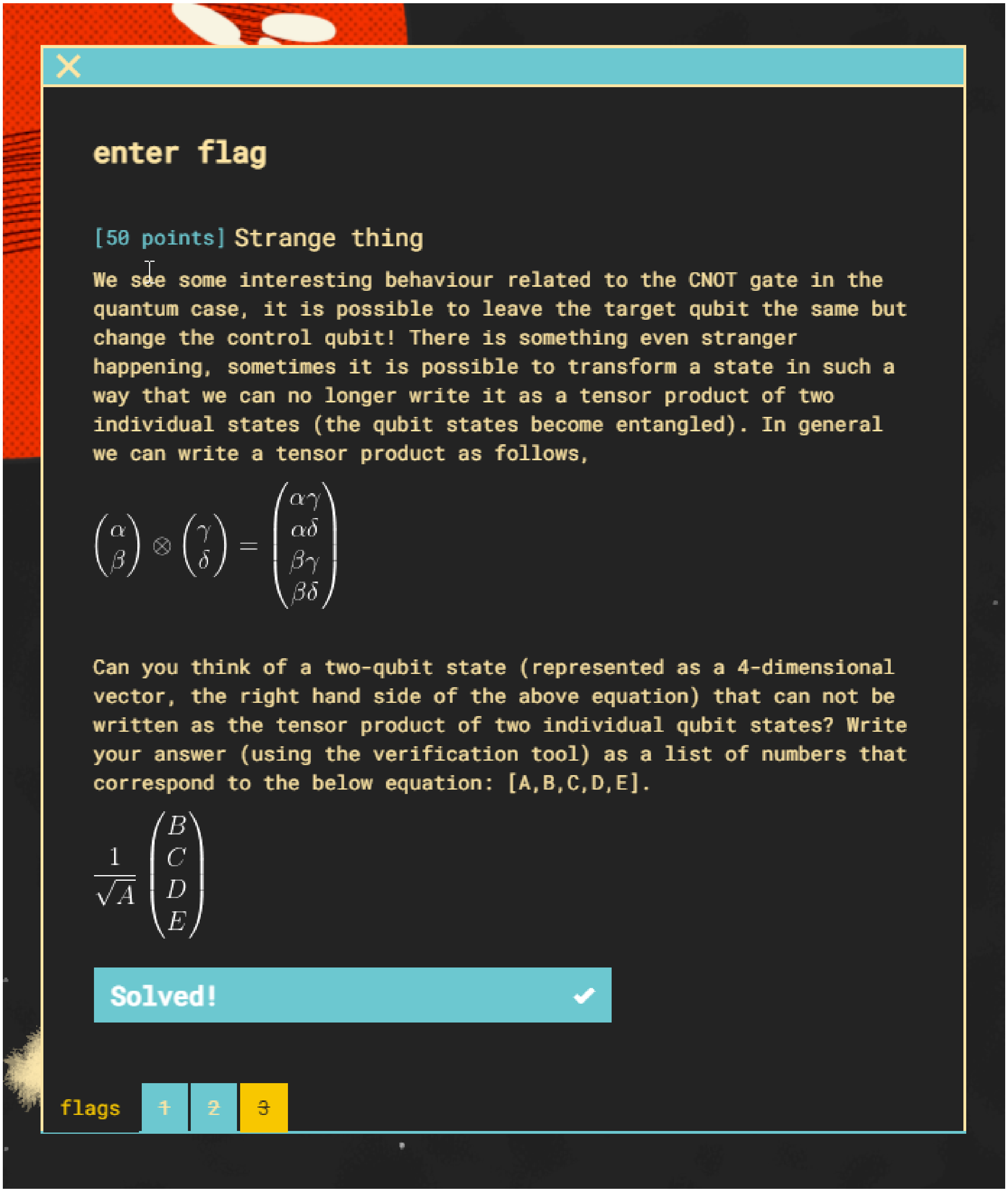
3> Strange thing

We see some interesting behaviour related to the CNOT gate in the quantum case, it is possible to leave the target qubit the same but change the control qubit! There is something even stranger happening, sometimes it is possible to transform a state in such a way that we can no longer write it as a tensor product of two individual states (the qubit states become entangled). In general we can write a tensor product as follows,

$$\begin{pmatrix} \alpha \\ \beta \end{pmatrix} \otimes \begin{pmatrix} \gamma \\ \delta \end{pmatrix} = \begin{pmatrix} \alpha\gamma \\ \alpha\delta \\ \beta\gamma \\ \beta\delta \end{pmatrix}$$

Can you think of a two-qubit state (represented as a 4-dimensional vector, the right hand side of the above equation) that can not be written as the tensor product of two individual qubit states? Write your answer (using the verification tool) as a list of numbers that correspond to the below equation: [A,B,C,D,E].

$$\frac{1}{\sqrt{A}} \begin{pmatrix} B \\ C \\ D \\ E \end{pmatrix}$$



enter flag

[50 points] Strange thing

We see some interesting behaviour related to the CNOT gate in the quantum case, it is possible to leave the target qubit the same but change the control qubit! There is something even stranger happening, sometimes it is possible to transform a state in such a way that we can no longer write it as a tensor product of two individual states (the qubit states become entangled). In general we can write a tensor product as follows,

$$\begin{pmatrix}\alpha\\ \beta\end{pmatrix} \otimes \begin{pmatrix}\gamma\\ \delta\end{pmatrix} = \begin{pmatrix}\alpha\gamma\\ \alpha\delta\\ \beta\gamma\\ \beta\delta\end{pmatrix}$$

Can you think of a two-qubit state (represented as a 4-dimensional vector, the right hand side of the above equation) that can not be written as the tensor product of two individual qubit states? Write your answer (using the verification tool) as a list of numbers that correspond to the below equation: [A,B,C,D,E].

$$\frac{1}{\sqrt{A}}\begin{pmatrix}B\\ C\\ D\\ E\end{pmatrix}$$

Solved!

flags ~~1~~ ~~2~~ ~~3~~

Points [50 points]

Solution

- We have to find out a exception to the qbit tensor products where a two-qubit state cannot be represented by the product of two individual qubit state after some trial and error of the tensor products I found out the the two-qubit state [$1/\sqrt{2}, 0, 0, 1/\sqrt{2}$] cannot be achieved by any combination of tensor products .
- When submitted this value to the validator I got the flag

Flag:- CTF{quantum_shooting_for_the_stars}

- We can also confirm that our calculation is correct from the given ref link below which also specifies the same state.

Reference Link:- https://docs.microsoft.com/en-us/azure/quantum/concepts-multiple-qubits