Hi!
Once again, thank you for your interest in the QC Mentorship program!

We decided to select participants based on how they will manage to do some simple “screening tasks”

These tasks have been designed to:
- find out if you have the skills necessary to succeed in our program.
- be doable with basic QC knowledge - nothing should be too hard for you to quickly learn.
- allow you to learn some interesting concepts of QC.
- give you some choices depending on your interests.

What we mean by skills is not knowledge and expertise in QC. It’s the ability to code,9 learn new concepts and to meet deadlines.

What are we looking for in these applications?
- Coding skills – clear, readable, well-structured code
- Communication – well-described results, easy to understand, tidy.
- Reliability – submitted on time, all the points from the task description are met
- Research skills – asking good questions and answering them methodically

Also, feel free to be creative – once you finish the basic version of the task, you can expand it. Bonus questions provide just some ideas on how to expand a given topic.

Choose tasks based on your interests, don’t try to pick the easiest one.

You need to do only 1 task. Feel free to do all of them, it might be a good learning opportunity, but it won’t affect admissions to the program :)

So here are the tasks:you have an infinite number of qubits.

Task 1 find the largest number

You have two integers, either positive or negative, and the challenge is to generate a quantum algorithm that returns which is the larger number. Consider an appropriate number of qubits and explain why your proposal is valid for all kinds of numbers in case

```
def find_the_largest_number (int:number_1, int ,number_2):
    “””
 number_1 : integer value that is the first parameter to the function,
number_2 : integer value that is the second parameter to the function.
```

Return the largest number between number_1 and number_2
    “””

    # use a framework that works with quantum circuits, qiskit, cirq, pennylane, etc.

    # consider print your quantum circuit,b

Example:

A = find_the_largest_number(5,-6)
print(A)

“5”

References:

[1] Deutsch, David, and Richard Jozsa. "Rapid solution of problems by quantum computation." Proceedings of the Royal Society of London. Series A: Mathematical and Physical Sciences 439.1907 (1992): 553-558.

[2] Bernstein, Ethan, and Umesh Vazirani. "Quantum complexity theory." SIAM Journal on computing 26.5 (1997): 1411-1473.

[3] Grover, Lov K. , "A fast quantum mechanical algorithm for database search", Proceedings of the 28th Annual ACM Symposium on the Theory of Computing (1996), arXiv:quanet-ph/9605043

Task 2 Is Rectangle?

Given four positive integers A, B, C, D, determine if there’s a rectangle such that the lengths of its sides are A, B, C and D (in any order).

If any such rectangle exist return 1 else return 0.

def is_rectangle (int:A, int:B, int:C, int:D):
    “””
A : integer value that is one side of the rectangle.
B : integer value that is one side of the rectangle.

C : integer value that is one side of the rectangle.
D : integer value that is one side of the rectangle.
Return if is a rectangle return 1 else 0.
    “””

    # use a framework that works with quantum circuits, qiskit, cirq, pennylane, etc.

    # consider print your quantum circuit,

Example:

A = is_rectangle(5,6,6,5)
print(A)

“1”

Task 3 QSVM

Generate a Quantum Support Vector Machine (QSVM) using the iris dataset and try to propose a kernel from a parametric quantum circuit to classify the three classes(setosa, versicolor, virginica) using the one-vs-all format, the kernel only works as binary classification. Identify the proposal with the lowest number of qubits and depth to obtain higher accuracy. You can use the $UU^{\dagger}$ format or using the Swap-Test.

Task 4 Random Circuits

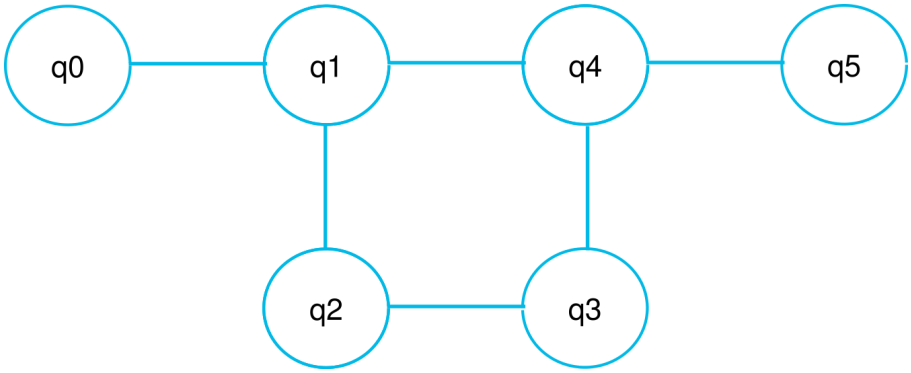
Design a function that generates a random quantum circuit by considering as parameters the number of qubits, the number of depths, and the base of gates to be used. You could only use the quantum gates of 1 and 2 qubits.

def random_circuit (int:num_qubits, int:depth, list:basis_gates):
    “””
num_qubits : integer value that is the number of qubits.
depth: integer value that is the depth for the random circuit.
basis_gates : A list that contains the basis gates to generate the quantum circuit.
Return the quantum circuit
    “””

use a framework that works with quantum circuits, qiskit, cirq, pennylane, etc.

print your quantum circuit,

Bonus: use the following order between qubits of a quantum computer



Deadline
2 weeks from when you've submitted your application in your timezone.
This means that if you submitted your application on February 15th, you can send your solution by midnight of March 1st.

Once you have finished a screening task, please submit your GitHub repository containing the code to this google form: https://forms.gle/Lkh1J9JvnN34yMY99 -- other forms of submission will not be accepted!

If you have any questions - please add comments to this document, or ask it in the QOSF slack workspace (invitation link) in the #mentorship-applicants channel. We will be updating this document with more details and/FAQ to avoid confusion, so make sure to check it before asking :)

Have a nice day!
QOSF team

FAQ

Q: Can we use any quantum libraries or are we restricted to a particular set of tools?

A: Feel free to use whatever you like, just make sure that the tool doesn't solve the whole problem for you.

Regarding the language of choice, Python is definitely the preferred one, since this is the language that most of the mentors use.
You can do the task first in the language of your preference and then translate it to Python if that's more convenient for you.

Q: I am applying as a member of a team. How many tasks do we submit?

A: Each member of a team must submit their own screening task. This will help us judge the skill level of each individual team member and help us pair folks up with the right mentor.

Q: How should I submit the solution?

A: All the materials for the submission should be inside a GitHub repository. Please do not send us any loose files as attachments or in any other format. Please submit your GitHub repository to this google form once you've finished: https://forms.gle/fmAYLVSUMGbjxK7x8
Q: My team-mate wants to leave the team because he/she/they can't manage these along with exams. So will this affect our team status or anything like that?

A: Well, just let us know and you can continue as an individual/smaller team.

Q: Is it possible to make more than one task and send everything together?
A: Yes, you can. But you should specify which task you want to be evaluated. In other words, do it as an exercise but it does not affect your chances to enter the program.

Q: Can I please get the slack link? I think the link has expired ?

A: try this: https://qosf.slack.com/archives/C019UEZRCM9
Another one to try:
https://join.slack.com/t/qosf/shared_invite/zt-vb1ftjp1-D2gpVKEfl6Ifv9oXvk_xDw