YQuantum 2025 - Super Hash Function Challenge

> *"Quantum computations are the next frontier in securing blockchain with advanced proof-of-work." – Anonymous, probably*

Welcome to Superquantum challenge at **YQuantum 2025**! This repository houses resources and examples for developing a quantum-based hash function, as described in our challenge prompt. Below you'll find instructions on how to use and navigate the materials, as well as details on submission and evaluation.

YQuantum 2025 - Défi de la Super Fonction de Hachage

> *"Les calculs quantiques sont la prochaine frontière pour sécuriser la blockchain avec une preuve de travail avancée." – Anonyme, probablement*

Bienvenue au défi Superquantum de **YQuantum 2025** ! Ce dépôt contient des ressources et des exemples pour développer une fonction de hachage quantique, comme décrit dans notre énoncé de défi. Vous trouverez ci-dessous des instructions sur la façon d'utiliser et de naviguer dans les matériaux, ainsi que des détails sur la soumission et l'évaluation.

Contents / Contenu

Challenge Description / Description du Défi

The heart of this repository is the **challenge.md** file. It details the goal of the challenge and provides the necessary information for you to understand the problem.

Le cœur de ce dépôt est le fichier **challenge.md**. Il détaille l'objectif du défi et fournit les informations nécessaires pour comprendre le problème.

Example Notebook / Notebooks d'Exemple

This repository provides an example Jupyter notebook **example.ipynb**. It provides an example quantum hash function and analyzes it based on our judging criteria.

Ce dépôt fournit un notebook Jupyter d'exemple **example.ipynb**. Il propose une fonction de hachage quantique et l'analyse selon nos critères d'évaluation.

Qhash Implementation / Implémentation de Qhash

We also provide a simplified implementation of the Qubitcoin's hash algorithm in the **qhash.py** (which we internally call qhash) for you to analyze. This implementation lacks the post-simulation classical hashing present in the blockchain to make it more in line with the challenge requirements. However, it only accepts 256-bit inputs, which would make it not eligible as a challenge solution.

Nous fournissons également une implémentation simplifiée de l'algorithme de hachage de Qubitcoin dans le fichier **qhash.py** (que nous appelons en interne qhash) pour que vous puissiez l'analyser. Cette implémentation ne contient pas le hachage classique post-simulation présent dans la blockchain, afin de mieux répondre aux exigences du défi. Cependant, elle n'accepte que des entrées de 256 bits, ce qui la rend inéligible comme solution au défi.

Recommended Environment & Dependencies / Environnement Recommandé et Dépendances

`example.ipynb` and `qhash.py` require the following dependencies to be run locally:

- qiskit
- numpy

`example.ipynb` et `qhash.py` nécessitent les dépendances suivantes pour être exécutés localement :

- qiskit
- numpy

Documentation & Write-up / Documentation et Rapport

A well-documented solution is key:

- **Code Documentation**: Add docstrings and inline comments explaining your logic, especially where the quantum portion is crucial (i.e., the “hashing” circuit).
- **Write-up**: Provide a `writeup.pdf` (or an equivalent Markdown/LaTeX file) detailing your approach:
    - Explanation of your circuit design.
    - Performance and quality analysis of your output.
    - Rationale for how your function meets the challenge requirements.

    Optionally:

- Analysis of the Qubitcoin's hash algorithm.

Une solution bien documentée est essentielle :

- **Documentation du Code** : Ajoutez des docstrings et des commentaires en ligne expliquant votre logique, en particulier là où la partie quantique est cruciale (c'est-à-dire le circuit de "hachage").
- **Rapport** : Fournissez un `writeup.pdf` (ou un fichier Markdown/LaTeX équivalent) détaillant votre approche :
  - Explication de la conception de votre circuit.
  - Analyse des performances et de la qualité de votre sortie.
  - Justification de la manière dont votre fonction répond aux exigences du défi.

  Optionnellement :

  - Analyse de l'algorithme de hachage de Qubitcoin.

Submission Guidelines / Directives de Soumission

To submit your final project:

1. **Include Source Code**:
   Place your core hashing function in the `main.py` that takes `bytes` as input and returns `bytes` as output.
2. **Include Documentation**:
   Provide a `writeup.pdf` summarizing your design, plus a brief presentation (`presentation.pptx`).
3. **Include Examples**:
   Demonstrate, in a separate notebook or Python script, how you tested your hashing function's performance (time, uniformity, etc.).

Pour soumettre votre projet final :

1. **Inclure le Code Source** :
   Placez votre fonction de hachage principale dans le fichier `main.py`, qui prend des `bytes` en entrée et retourne des `bytes` en sortie.
2. **Inclure la Documentation** :
   Fournissez un `writeup.pdf` résumant votre conception, ainsi qu'une brève présentation (`presentation.pptx`).
3. **Inclure des Exemples** :
   Démontrez, dans un notebook ou un script Python séparé, comment vous avez testé les performances de votre fonction de hachage (temps, uniformité, etc.).

Evaluation Criteria / Critères d'Évaluation

Submissions will be judged according to the criteria outlined in `challenge.md`:

1. **Output determinism**
2. **Entropy Preservation**
3. **Computational Difficulty**
4. **Preimage & Collision Resistance**
5. **Feasibility** – Not exceeding 20 qubits for up to 256-bit inputs.
6. **Speed** – Reasonable execution times.
7. **Purely Quantum Hashing** – No offloading to classical hash functions.

Additional points (a lot of them) may be awarded for thorough proofs or analyses of your function, and the corresponding analysis of the Qubitcoin's hash algorithm.

Les soumissions seront évaluées selon les critères décrits dans `challenge.md` :

1. **Déterminisme de la sortie**
2. **Préservation de l'entropie**
3. **Difficulté de calcul**
4. **Résistance aux préimages et aux collisions**
5. **Faisabilité** – Ne pas dépasser 20 qubits pour des entrées allant jusqu'à 256 bits.
6. **Vitesse** – Temps d'exécution raisonnables.
7. **Hachage Purement Quantique** – Pas de recours à des fonctions de hachage classiques.

Des points supplémentaires (beaucoup) peuvent être attribués pour des preuves ou analyses approfondies de votre fonction, ainsi que pour l'analyse correspondante de l'algorithme de hachage de Qubitcoin.

License & Attribution / Licence et Attribution

All files in this repository, including the notebooks and challenge materials, are distributed for educational purposes.

Tous les fichiers de ce dépôt, y compris les notebooks et les matériaux du défi, sont distribués à des fins éducatives.