



Квантове
програмування.
Справжні випадкові
числа

Доповідає Сапожник Дмитро

Познайомимося

- Мене звати Дмитро
- Займаюсь Фронтенд розробкою на роботі та всім не на роботі :-)
- Аспірант на кафедрі ІПЗ в Житомирській Політехніці



План

01.

Поговоримо про квантове програмування, квантову інформацію та квантову фізику в цілому, зрозумілими словами.

02.

Випадкові числа в JavaScript особливості та застосування(Math.random).

Випадкові числа у квантовому програмуванні. Реалізуємо програму для генерації випадкового біту та числа у діапазоні.

03.

Розкажу і покажу як почати вивчати квантове програмування. Поділюся корисними посиланнями.

Закон мура

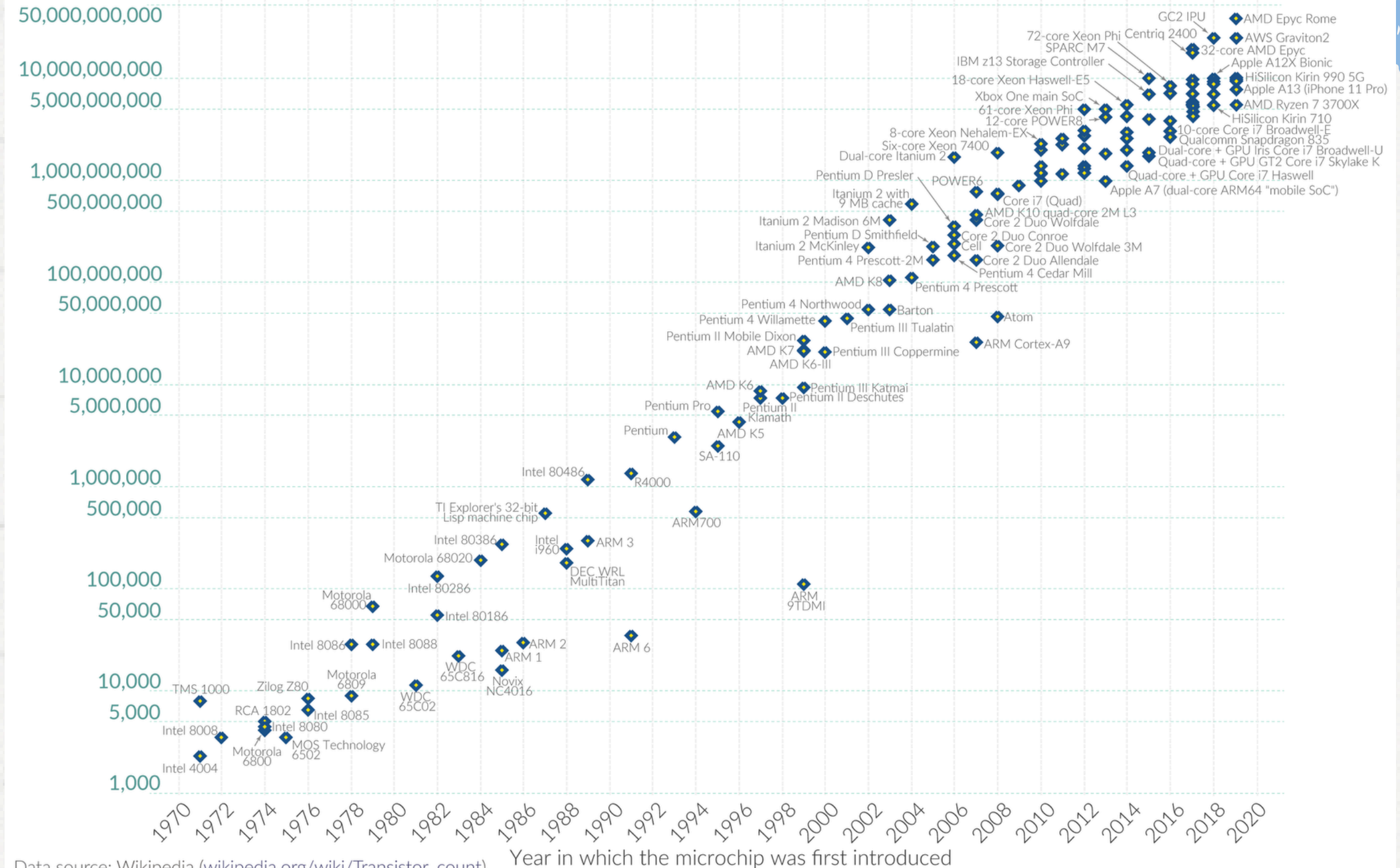
Кількість транзисторів на кристалі мікросхеми подвоюватиметься кожні 24 місяці

Moore's Law: The number of transistors on microchips doubles every two years

Moore's law describes the empirical regularity that the number of transistors on integrated circuits doubles approximately every two years. This advancement is important for other aspects of technological progress in computing – such as processing speed or the price of computers.



Transistor count



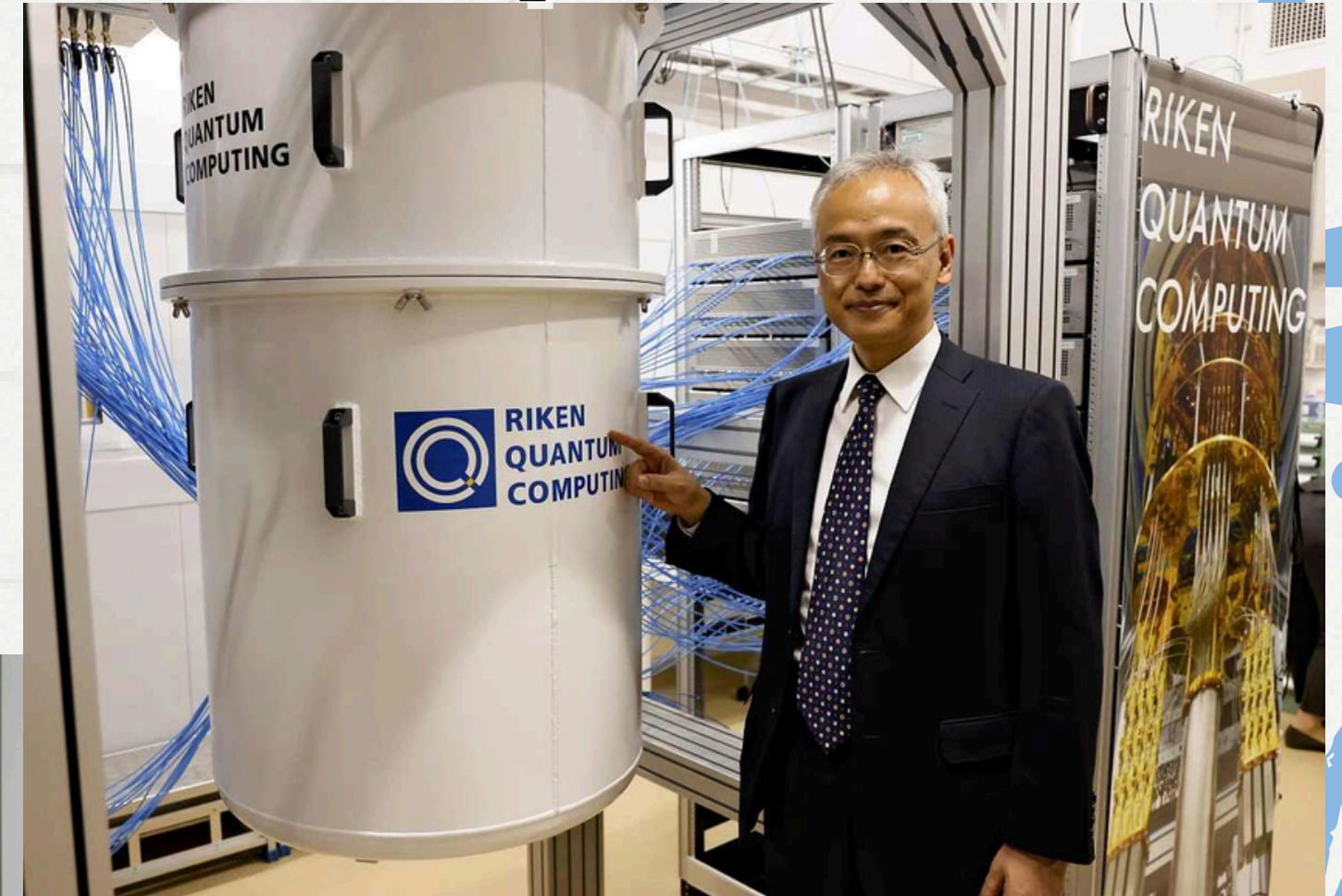
Data source: Wikipedia (wikipedia.org/wiki/Transistor_count) OurWorldinData.org – Research and data to make progress against the world's largest problems. Licensed under CC-BY by the authors Hannah Ritchie and Max Roser.

Перша думка про квантовий комп'ютер

1981 р. – “Для
вирішення задач
квантової фізики нам
потрібний квантовий
комп'ютер”



Як виглядає квантовий комп'ютер



Гібридні обчислення



Виконує звичайні
обчислення

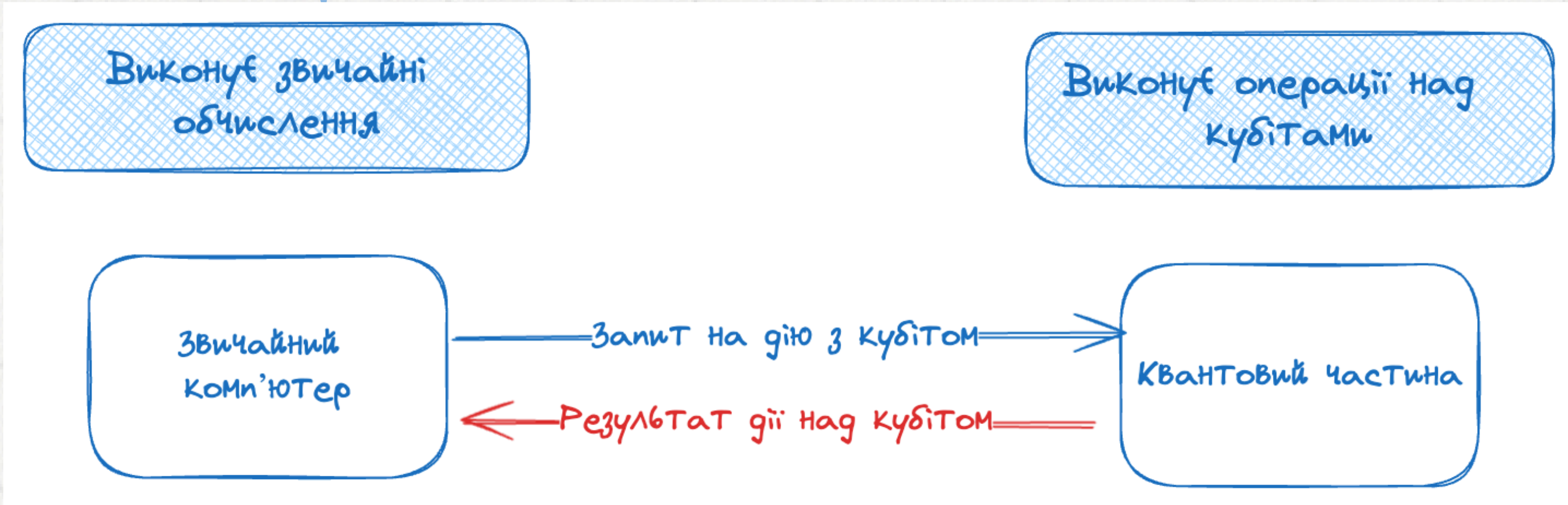
Виконує операції над
кубітами

Звичайний
комп'ютер

Запит на гіт з кубітом

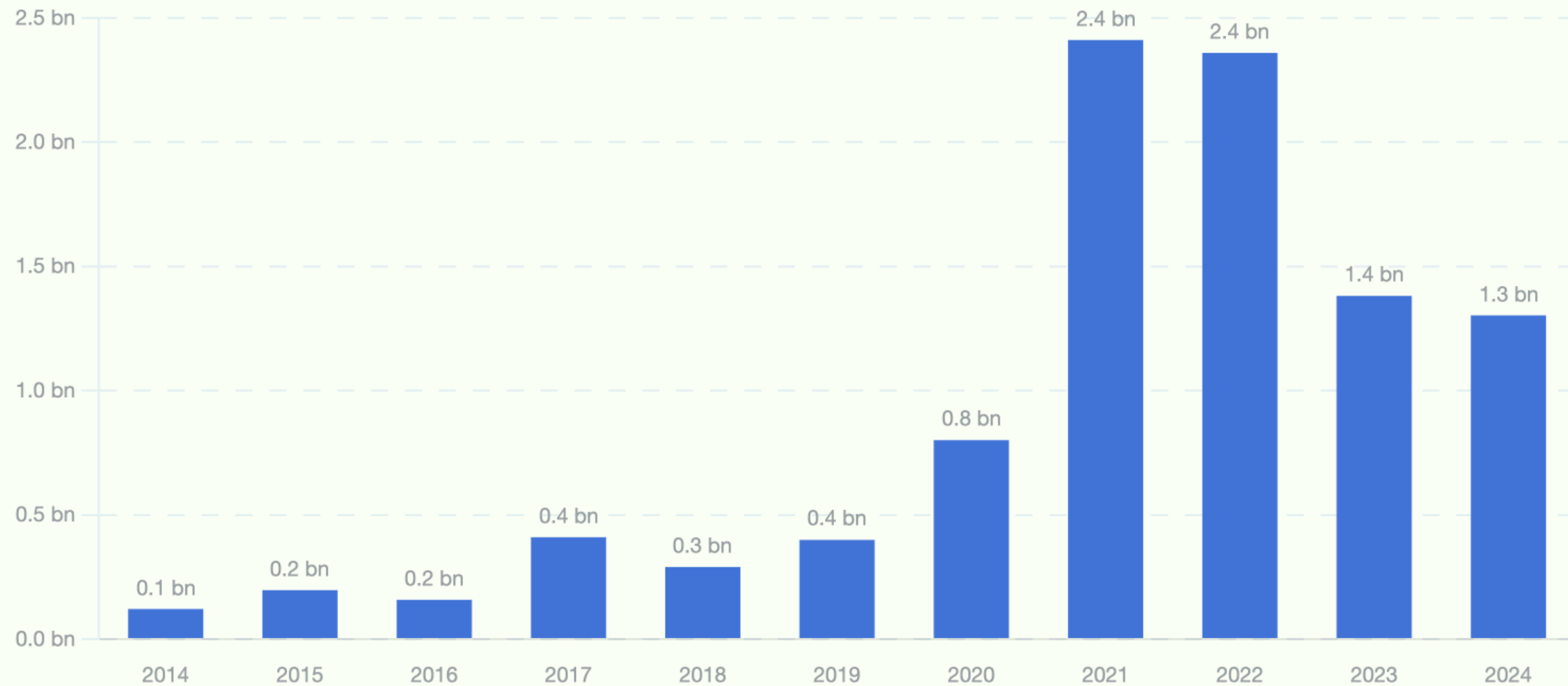
Квантовий частина

Результат гіт над кубітом



Інвестиції

Quantum Tech Investment (USD)



Компанії які займаються КВАНТОВИМИ ТЕХНОЛОГІЯМИ



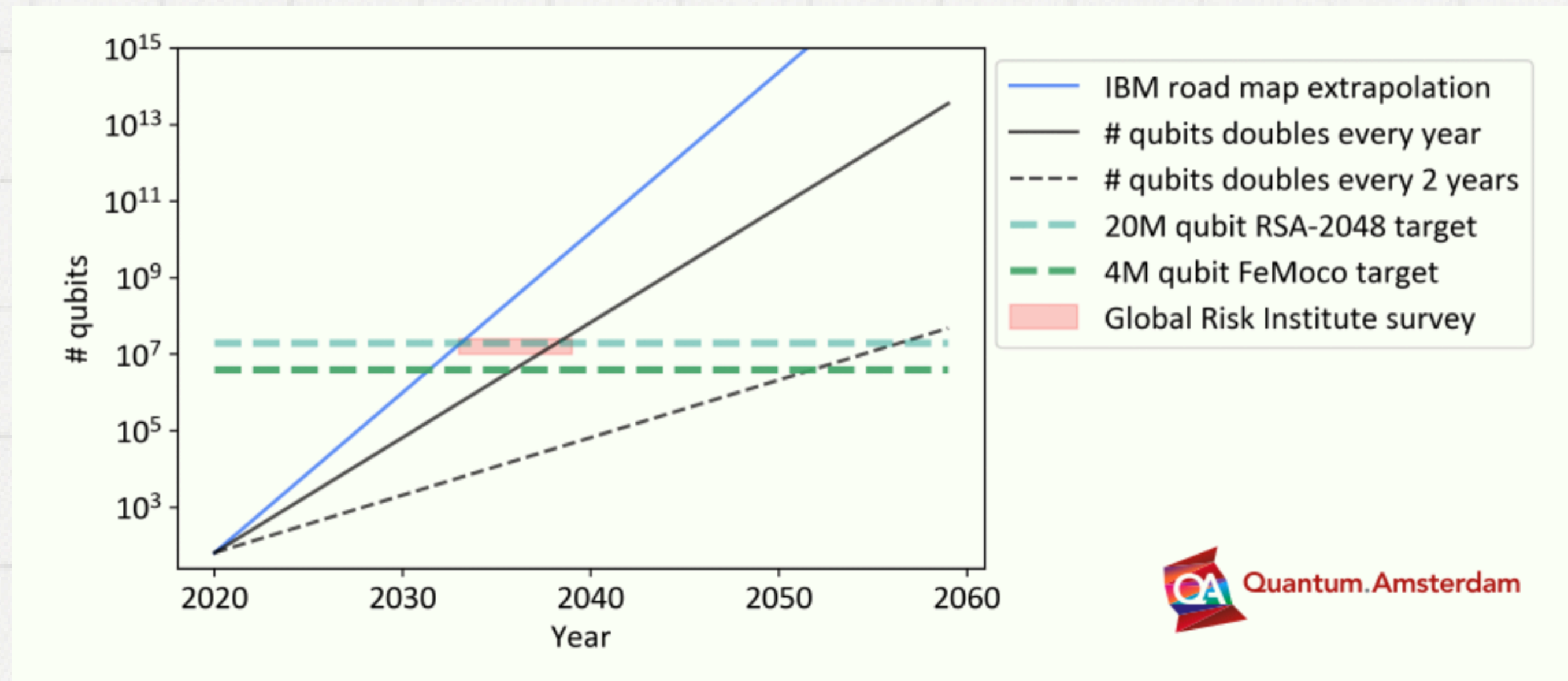
softserve

D:WAVE



ALICE & BOB

Перспективи КІЛЬКСТІ КУБІТІВ





Квантова фізика

Квантова фізика — розділ теоретичної фізики, в якому вивчаються квантово-механічні та квантово-польові системи і закони їх руху.

Кіт шредінгера

Живий чи мертвий? Дізнаємось тільки тоді коли відкриємо коробку.



Класичний біт

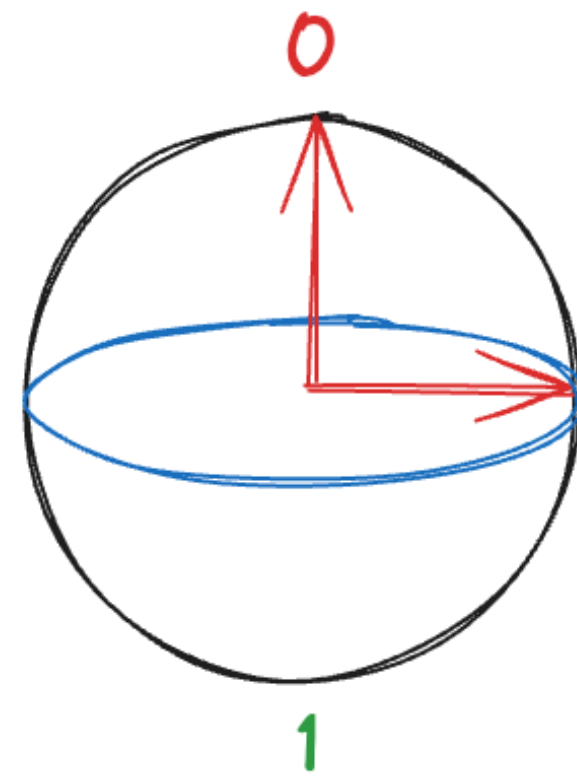


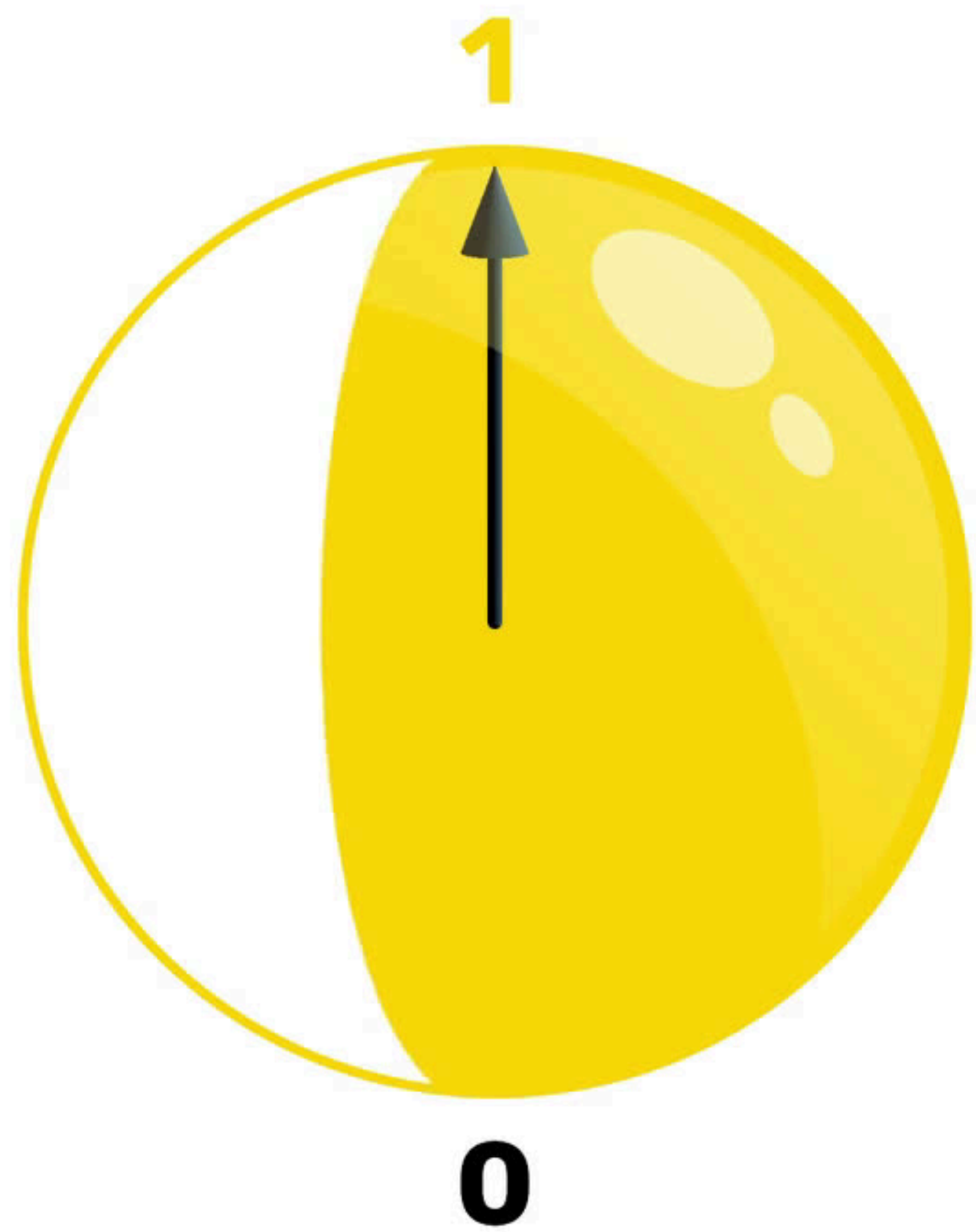
0



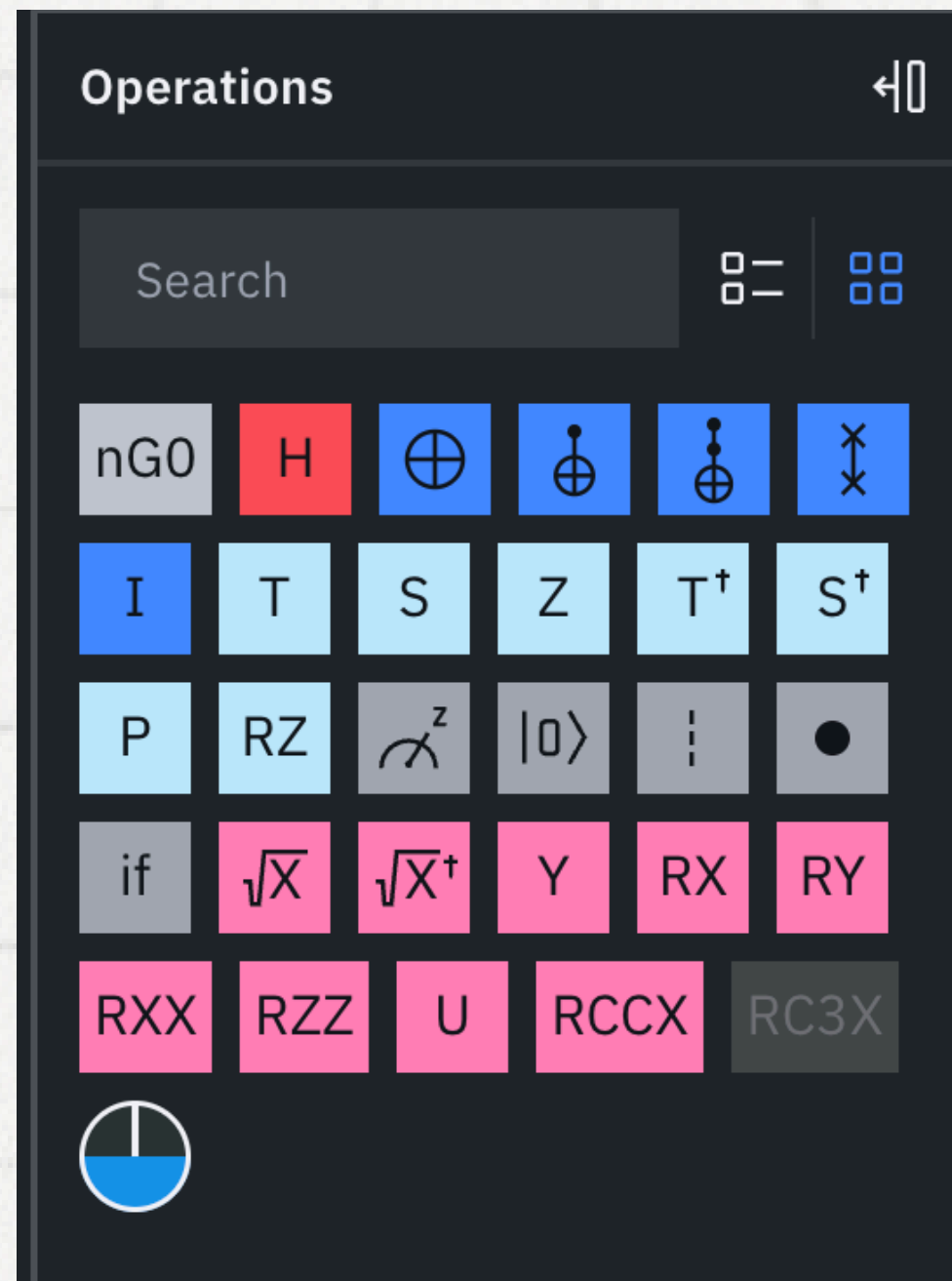
1

Квантовий біт





Гейти – управління кубітом



Класичний AND



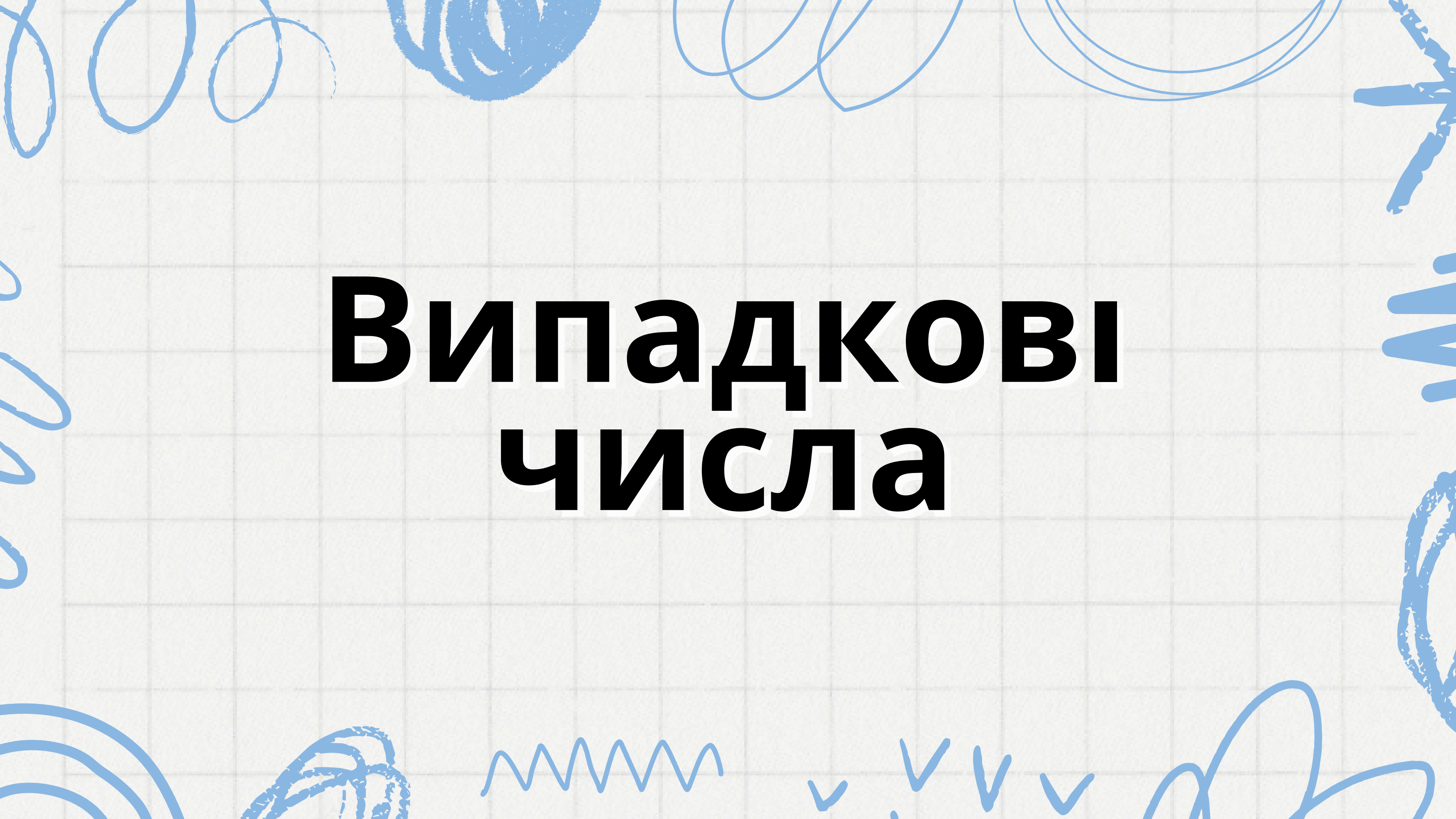
Input 1	Input 2	Output
0	0	0
0	1	0
1	0	0
1	1	1

Гейт Адамара



Матриця перетворення

$$\hat{H} = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}$$



Випадкові числа

Як працює Math.random() в JS

Декілька моментів які хочеться виділити:

1. Math.random() - це частина браузерного API, не частина мови
2. У всіх браузерах після 2015 року для Math.random() використовується алгоритм генерації псевдовипадкових чисел xorshift128+
3. xorshift128+ - використовує сім фраз для встановлення початкових значень state0 та state1. Код можна алгоритму можна побачити нижче:

Copy

```
uint32_t state0 = 1;
uint32_t state1 = 2;
uint32_t mwc1616() {
    state0 = 18030 * (state0 & 0xFFFF) + (state0 >> 16);
    state1 = 30903 * (state1 & 0xFFFF) + (state1 >> 16);
    return state0 << 16 + (state1 & 0xFFFF);
}
```

```
$ node --random_seed=42
Welcome to Node.js v16.0.0-pre.
Type ".help" for more information.
> Math.random()
0.7939112874678715
> for (let i = 0; i < 1000; i++) Math.random();
0.6681221903420669
> Math.random()
0.009229884165582902
>
$ node --random_seed=42
Welcome to Node.js v16.0.0-pre.
Type ".help" for more information.
> Math.random()
0.7939112874678715
> for (let i = 0; i < 1000; i++) Math.random();
0.6681221903420669
> Math.random()
0.009229884165582902
```

The background features a light blue grid pattern. Scattered throughout are various hand-drawn blue scribbles, including loops, swirls, and zig-zags, giving it a creative and artistic feel.

Квантове програмування

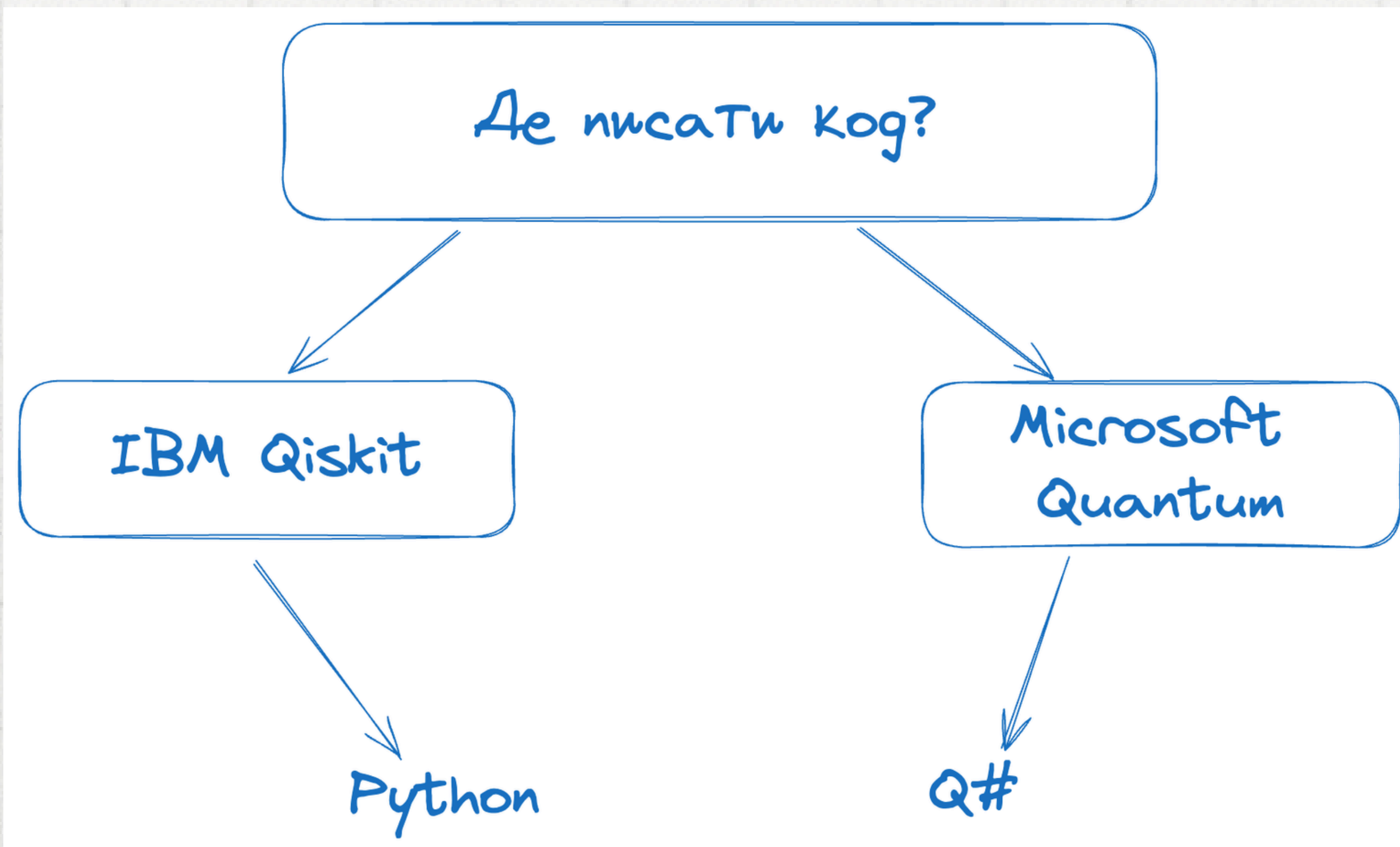
Де писати код?

IBM Qiskit

Microsoft
Quantum

Python

Q#



Q# – КОД

```
1 namespace HGateSample {  
2     @EntryPoint() // Позначення точки входу  
   Run | Histogram | Estimate | Debug | Circuit  
3     operation Main() : Result {  
4         use q = Qubit(); // Створення кубіта  
5         H(q); // Застосування гейта Адамара  
6  
7         let result = M(q); // Вимірювання кубіта  
8  
9         Reset(q); // Повернення кубіта у стан |0>  
10        return result; // Повернення результату вимірювання  
11    }  
12 } ✨
```

Q# – результат

Result: "Zero"

Finished shot 1 of 1

Q# simulation completed.

IBM Qiskit

Untitled circuit *Saved* File Edit View

Operations Left alignment Inspect

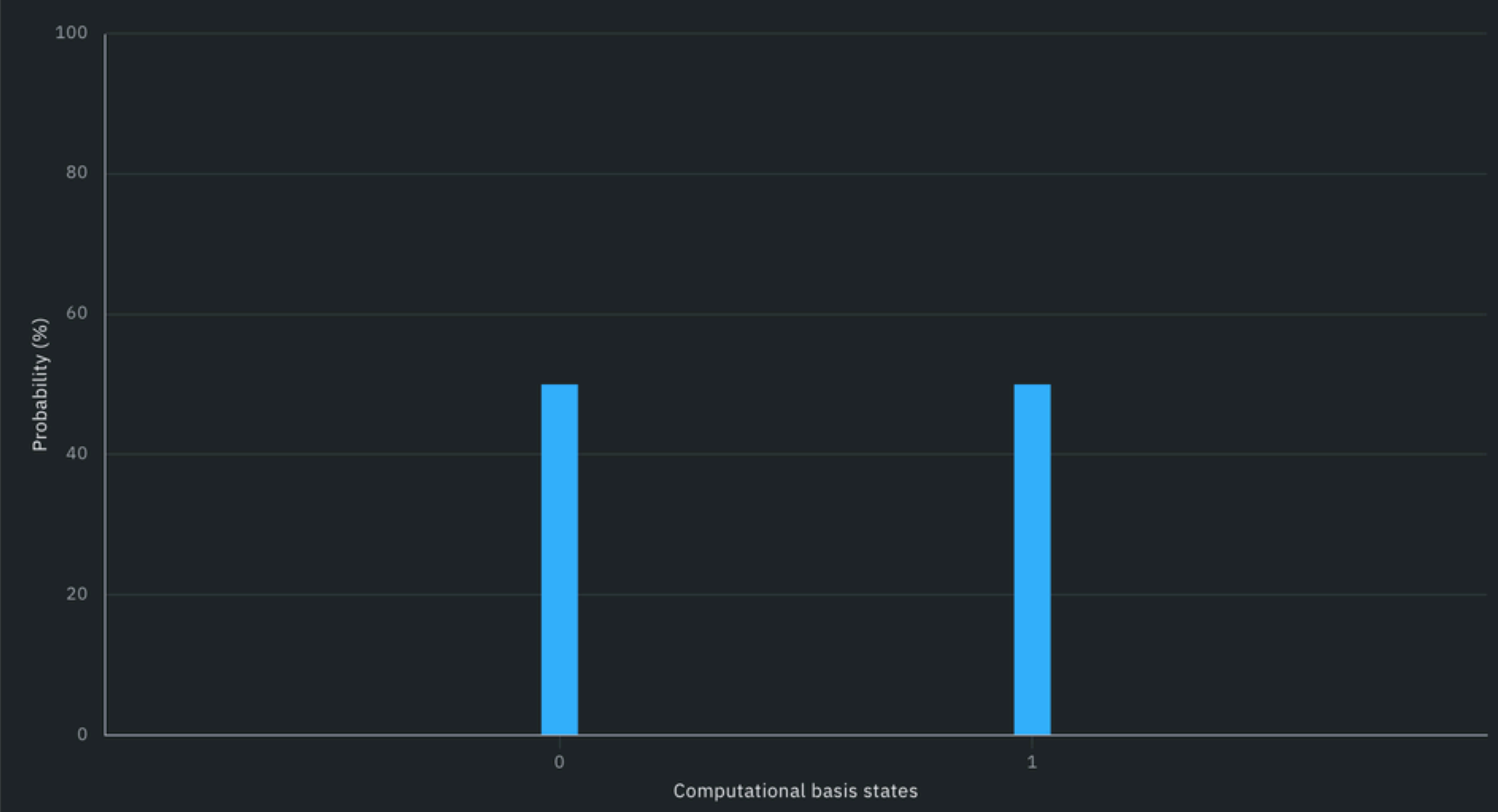
Search

q[0] **H**

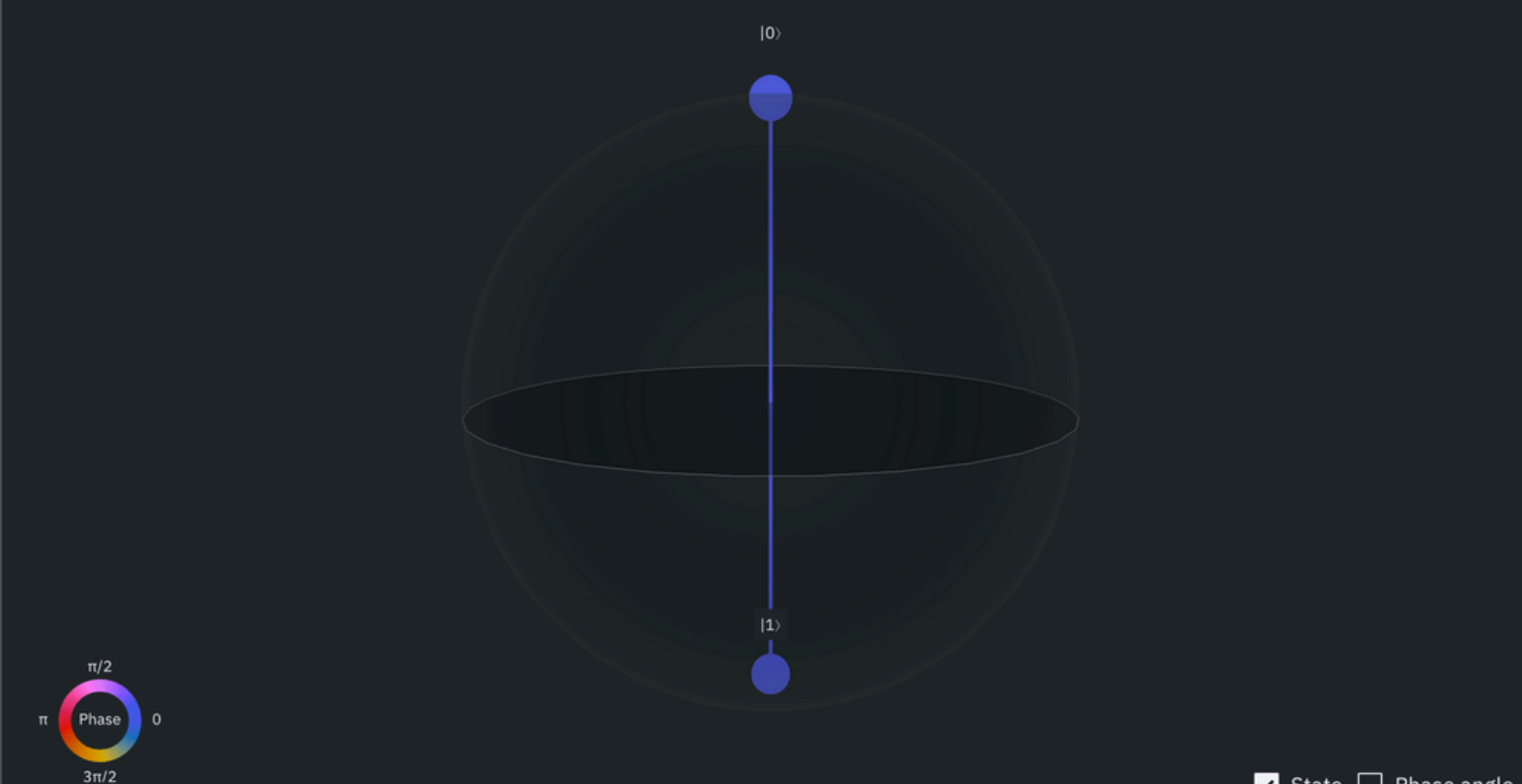
c2

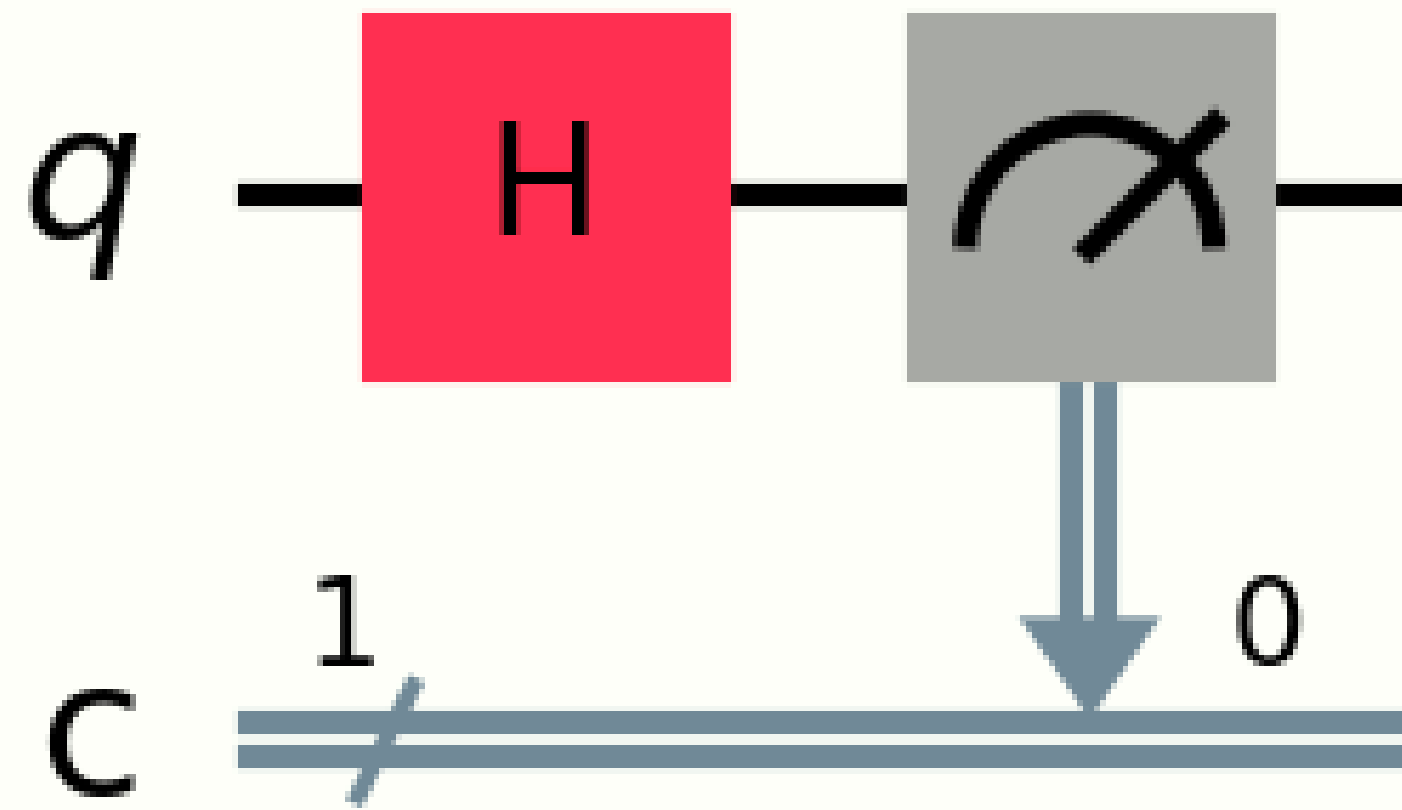
nG0 **H** \oplus \oplus \oplus \otimes
I **T** **S** **Z** **T[†]** **S[†]**
P **RZ** \curvearrowright $|0\rangle$ $|1\rangle$ \bullet
if \sqrt{X} \sqrt{X}^\dagger **Y** **RX** **RY**
RXX **RZZ** **U** **RCCX** **RC3X**

Probabilities ⓘ ⋮



Q-sphere ↺ ⓘ ⋮






```
from qiskit.visualization import plot_histogram
from qiskit import QuantumCircuit
from qiskit_aer import AerSimulator

sim = AerSimulator()

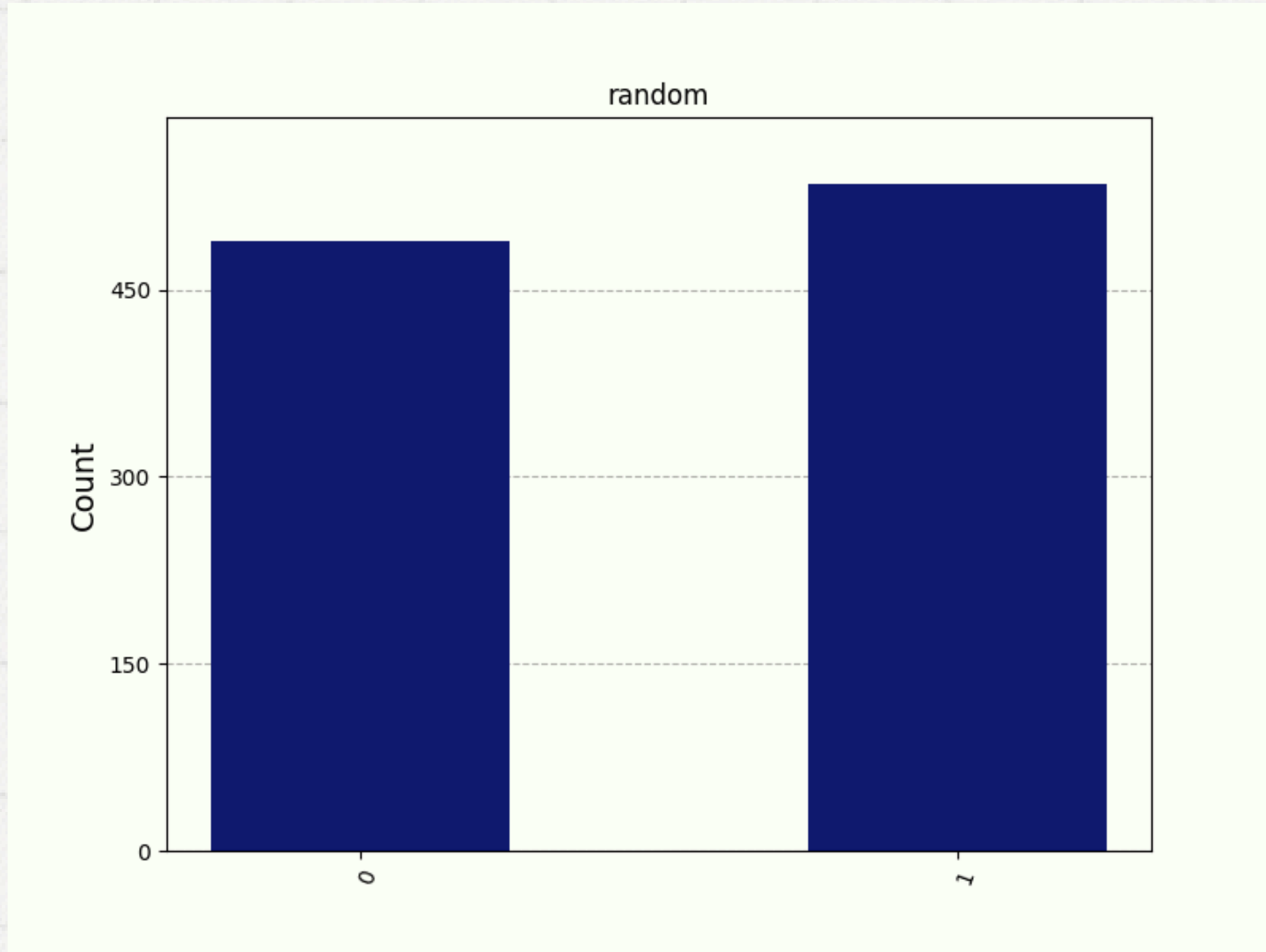
qc = QuantumCircuit(1, 1)

qc.h(0)
qc.measure(0, 0)

qc.draw(output='mpl', filename='random.png')

statistics = sim.run(qc).result().get_counts()

histogram = plot_histogram(statistics, title='random', bar_labels=False, color='midnightblue', figsize=(8, 6), filename='result.png')
```



Ресурси

IBM Learn

Фундаментальні речі про квантову інформацію та квантові алгоритми

q-ctrl.com

Інтерактивна платформа для навчання

Microsoft Learn

Корисна інформація про квантове програмування та саме мову Q#

d-wave leap

Корисна інформація про квантове програмування, інтерактивні дета приклади коду

**Дякую за
увагу**

www.iot-sdn.space