Lab 2 — SAC & BIC for Cryptographic Hash Functions + Collisions

Goal

Measure and visualize **SAC** (Strict Avalanche Criterion) and **BIC** (Bit Independence Criterion) for a strong hash (e.g., SHA-256) and for a simple toy hash; understand why good hashes show ~50% avalanche and near-zero inter-bit correlations. Explore collisions/near-collisions in a toy hash.

Learning outcomes

- Compute and interpret **SAC matrices** and **BIC correlations**.
- Explain why strong hashes exhibit ~0.5 flip probability per output bit and near-zero pairwise dependencies.
- Compare naive search vs. simple heuristics for (near-)collision finding in a weak hash.

Setup

- Use Python (or similar) with basic numerics/plotting.
- Fix a **constant message length** for all experiments (to avoid padding artifacts).

Part A — SAC (Strict Avalanche Criterion)

1. **Parameters**
    - Number of random messages: start with **N = 2,000**, then (time permitting) **N = 10,000**.
    - Message length (bytes): fixed for all trials.
    - Input bits tested: at least **32 randomly chosen** input bits (full set optional).
2. **Procedure**
    - For each tested input bit (i): a) Sample a random message (x). b) Compute (y = H(x)) and (y' = H(x \ ⊕ e_i)). c) Record the output-bit flip vector (\Delta = y \ ⊕y').
    - Aggregate over (N) trials to estimate (p_{i \to j} = \Pr[\Delta_j=1]) for each output bit (j).
    - This yields an **SAC matrix** of shape *(#input bits tested) × (#output bits)*.
3. **What to report**
    - Heatmap of the SAC matrix.
    - Per-output-bit averages and deviations.
    - Interpretation: strong hash ≈ uniform around 0.5; toy hash should show visible deviations/patterns.

Part B — BIC (Bit Independence Criterion)

Parameters

- Choose one or a few input bits (i) (from Part A).
- Use the same (N) and message length.

Procedure

- For each output bit pair ((j,k)), compute correlation between their flip indicators (\Delta y_j, \Delta y_k) (collected over the (N) trials with input bit (i) flipped).
- Summarize correlations (e.g., Pearson/phi) into a **BIC matrix** over selected output bits.

What to report

- Heatmap of inter-bit correlations.
- Interpretation: strong hash ≈ correlations near 0; toy hash may show clusters/dependencies.

Part C — Extended Task: (Near-)Collisions

- **Strong hash (truncated SHA-256 to (n) bits)**: attempt to find **any collision**. Compare to the birthday bound baseline; explain why ML/heuristics should **not** outperform random search here.
- **Toy hash (weak design)**: attempt a **collision** or **near-collision** (minimize Hamming distance of outputs). Compare:
- Baseline (random sampling / simple hill-climb) vs. a simple heuristic (e.g., evolutionary search or bandit-style bit flips).
- Report evaluations-to-success / convergence trend and discuss why the weak hash is exploitable.

Practical tips

- Keep message length **constant** across trials.
- Use vectorized operations for bit extraction/XOR and aggregation.
- Clearly separate **measurement (SAC/BIC)** from **interpretation**; avoid over-fitting to small (N).