Brandon's Cryptographically Secure Pseudorandom Number Generator

Time limit: 39.0s **Memory limit:** 1G

blin00 is trapped in Japan! His website is down, and in order to bring it back up he needs to ssh into his server. Because **blin00** is extremely security-conscious, he has enterprise-grade 2FA enabled. His security token is a hardware token that he has locally which will generate a random integer uniformly at random within the range configured on the token.

blin00 has logged the last 25K outputs of his security token and suspects that his token does not generate cryptographically secure pseudorandom numbers. Can you prove this?

Your job is to write a program to print the following file. Your score will be inversely correlated to the length of your program.

To ensure you have the correct output, the following hashes have been provided:

- MD5: `f5906769047253b2b465332c9c491301`
- SHA1: `092b15980480149069eac35b3334b870e1217830`
- SHA256: `490419e8ce605dfcd92a69b3b7da153b73ba3db7e1a6a363bd7a4637e456f47c`
- SHA512: `07c13e901a2b4036673ec821fb2fe2231827707044331d23c43b6ad33bb2a23c8496af2397f9691770d8d2d6c01bbc6283d476a8412b413726ffc9423851f825`
- SHA512/256: `87b447dbe7308692e35ae50d0d81a0ffcc3e5affbb3626fd8d52d04bbb00fa4a`
- SHA3-256: `b97396381775fcb8facfa9ff0c8c1de13b50300571d51a34eddc5131b6978529`
- SHA3-512: `c3a1bcb6a4826e7ee8dd576a5346efbb6d875559b8412109e06d3532e1afda248a2d50a4617abb9f99ba7929b67f4f733c538050cb3ef972cef249b97ff47478`