SSS

The description "Shamir will be proud" suggests Shamir's secret sharing.
The decoded strings seems to be reversed so we need to reverse it.
Decode script:

```
from binascii import unhexlify

P1 =
"8010ba0d6ed38ef563074c3ee80a44f7fe680e82015a8d35f7f2245f66ec9c889b4e31a0c3e
97bceeb6f28695f7a494918e0ca079677f07fff8eb570c17a4cb1db0477b84e9c68b9f02b21b
33850f33bbd18f886b65c1f3bb015ddbe2723e64abfe8595e181d69d3f8ca3b7cc01c875ea25
b97ef1e171c4f3f887e5752541270ae461cc610b3eb422c34df84e7b9a567f7933ee4b6969d1
9273d212a3ee92f8509679a4b40b6823c007e6d5c6241959e86bc8f989754649cd3008bdbb5b
f030c9e802adf54d3afce4edef9bb709c7db4c2ac1f96f3e05cd220534b5647f35888e0e3d24
35abdb1d7f32413bb630b3e8b0502e774dda8ac2bd4c2623ac433f79bd12"
P2 =
"80264e325aa037314746964303cf6fee98d64e1e03d613fb8f327f5241850adbd06e1f959bd
b6e5bd35874188e3fa4740a1948befcacb8949350574825ba4519793a6a617048fb2f5bdd9bc
3267a61051484ec16e83ff7baaafac81a3aa4fb2077da312ee4f00c705b8f626334ff3045e41
f451858988a3549e314f8a70f0879f5a30fbcd5fcc1645575186af8a434876304bb1ebc36053
3389143f7d918682307736bac713b63338482ef1cf80ac415f213625231ef3d3bdd70f811c8c
c7515cf83a74ea25c31264a9a5dbe0615c5959e181bf8effa1698ece11cb5e9c794d381311ba
1900f0c550f33b61fd49959d9b4ba73588b14906fddb625bd13f7149a95a"
P3 =
"8036f43f3473b9c42441da7debc52b1966be409c028c9ece78c05b0d276996534b202e35583
2159538375c71d145ed3d12f982b96adb48eb6cdee238e4c009a8a23e1dd93ed49396abf6ba7
01e2a923ea99c14905e63e8811aef15a41d871d6ac8326870fced65a3a345591ff4e3b71b464
4d2f7468f966a71bb698ff6cb198958d5105ac65f2c367a31c4fe1c0d97b09717867a09209e0
a1cac64ede1c144d60854f7c7321de22f82ec8470991b57db729feb8aa0eb5ab7081070aa7b3
3755952238b81f5cf9dc80724a26575d9bba15ae4027e1f9a490acfd25183adb4ca1b62a2ca9
2c9a2bee2fa27a634b4b26402b298975c509c3f240f344037d1a4e44142b"


def strip_prefix(hexstr: str) -> bytes:
    """Ensure even-length hex, decode to bytes, and drop the first two
bytes."""
    if len(hexstr) & 1:  # odd length → pad with '0'
        hexstr = "0" + hexstr
    return unhexlify(hexstr)[2:]


def xor_three(a: bytes, b: bytes, c: bytes) -> bytes:
    """XOR three byte arrays together."""
    return bytes(x ^ y ^ z for x, y, z in zip(a, b, c))
```

```
def decode_flag(p1: str, p2: str, p3: str) -> str:
    combined = xor_three(strip_prefix(p1), strip_prefix(p2),
strip_prefix(p3))
    return combined.decode("utf-16-be")[::-1].rstrip("\x00\x01")


if __name__ == "__main__":
    flag = decode_flag(P1, P2, P3)
    print(flag)
```