Hash

Problem Description

The Vanderbilt Blockchain Club needs help mining some VandyCoins to use as gifts for all the economically forward-thinking little girls and boys.

To do this, they need to find the MD5 hashes in which, in hexadecimal, start with at least **six zeroes**. The input to the MD5 has is some secret key, followed by a number in decimal. To mine VandyCoins, you must find the lowest positive number with no leading zeroes (1, 2, 3...) that produces such a hash.

For example, if the secret key is `abcdef`, the answer is **6742839** because the MD5 hash of `abcdef6742839` starts with 5 zeroes (`000000072a1e4320d13deee9d934ae29`), and is the lowest to do so.

Your input is `ckczppom`. Encapsulate your answer in `flag{}` braces.

Walkthrough

One of the first things that we notice for this problem is that it doesn't specify we need to use C. This is a blessing because high-level languages make this problem much easier. In this instance, we will use Python.

We need to perform MD5 hashing on a series of inputs. A quick Google search leads us to the `hashlib` library. This library has an `md5()` function that takes a byte string and gets the `md5()` hash of it.

We also find this GeeksForGeeks article that shows us exactly how to use it. Using the following sample code, we see how to get an MD5 hash:

```
import hashlib

result = hashlib.md5(b'GeeksforGeeks').hexdigest()
print(f"The byte equivalent of hash is : {result}")
```

This prints us the MD5 hash of the string in hexadecimal. This is perfect!

Now, we need to find the string such that the MD5 hash starts with six zeroes. MD5 is computationally expensive to reverse, but not to brute force. Therefore, we'll simply iterate across the values until we find one that works.

We can do this with the following code:

```
import hashlib

input = 'ckczppom'
key = 0
while True:
    string = input + str(key)
    hash = hashlib.md5(string.encode()).digest().hex()
```

```
    if hash[0:6] == '000000':
        print('flag{' + str(key) + '}')
        break
    key = key + 1
```

Running this takes a few seconds, but it gives us the flag: `flag{3938038}`.