Callable Solution

```
class Engine:
    def __init__(self):
        self.cache = {}
    @staticmethod
    def next100(x: float) -> int:
        return int(round(x, -2))
    def __call__(self, tach: float) -> int:
        t100 = self.next100(tach)
        if t100 not in self.cache:
            actual = self.next100(0.7724*t100**1.0134)
            self.cache[t100] = actual
        return self.cache[t100]
```

Caching Design Issue #1

- Exact Equality Tests and float values
- float values don't always match exactly
- They're a terrible choice for cache keys

```
>>> 100 + (1/3) - 100 == 1/3
False
>>> 100 + (1/3) - 100
0.3333333333333286
```