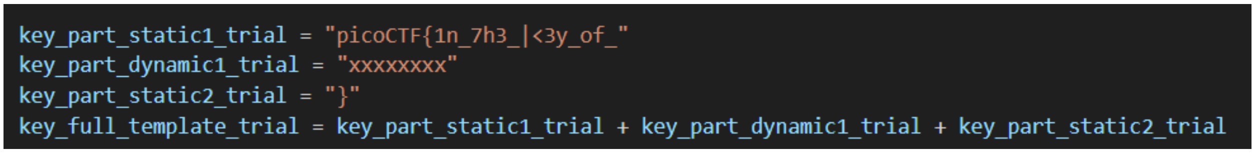
keygenme-py

Opened the file keygenme-trial.py that was given.
Tried to make sense of whatever the heck was going on in there.
It seemed to be a program where some space-thingy was going on.
After a few seconds of scrolling I saw this :

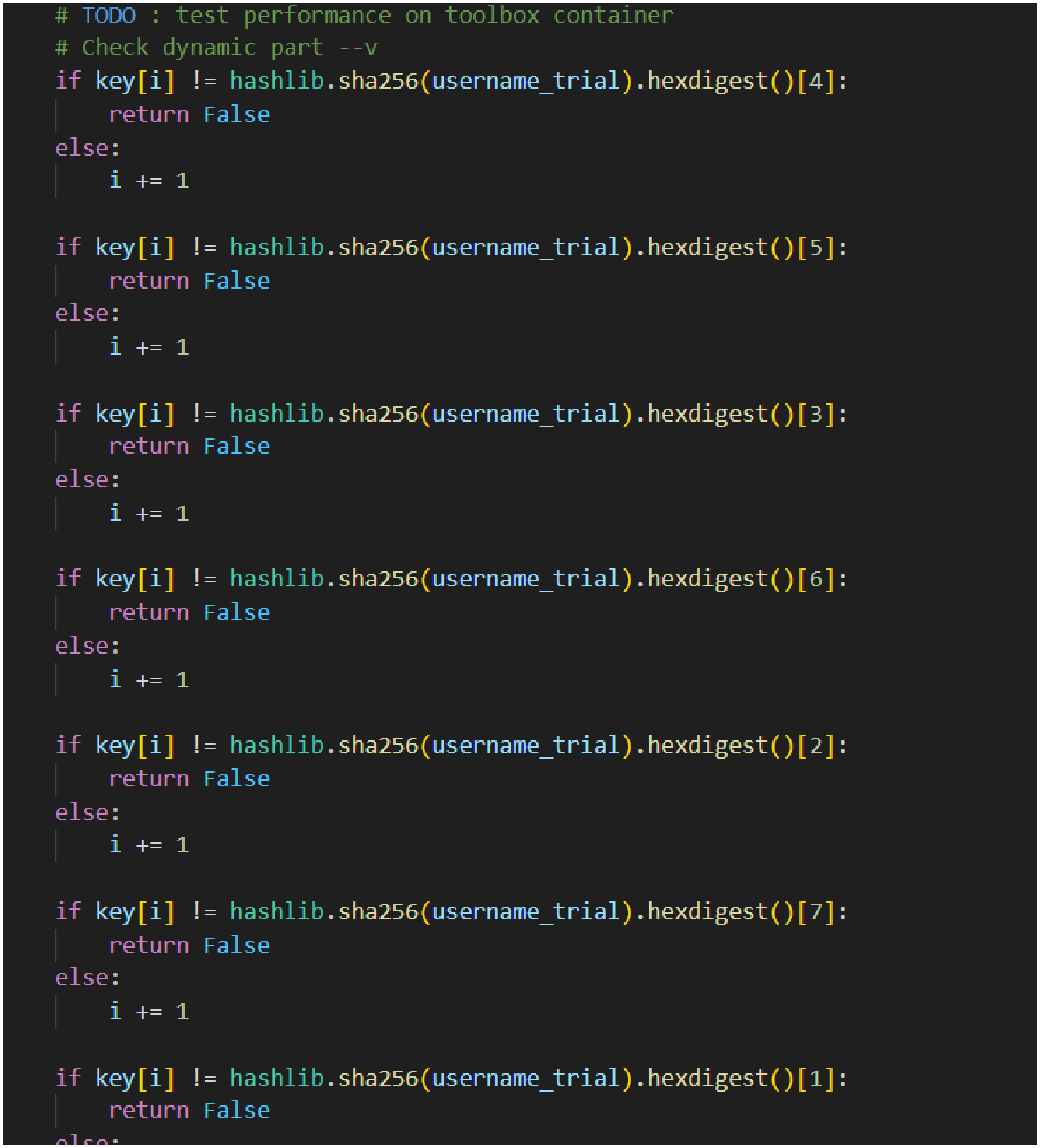


```
key_part_static1_trial = "picoCTF{1n_7h3_|<3y_of_"
key_part_dynamic1_trial = "xxxxxxxx"
key_part_static2_trial = "}"
key_full_template_trial = key_part_static1_trial + key_part_dynamic1_trial + key_part_static2_trial
```

Did attempt putting this, but of course.
That wasn't it

I noticed the "xxxxxxx" was being called the 'dynamic' part, unlike the other two that were 'static'.
So I figured something was prolly going on with the dynamic key part, and that's what I had to figure out.

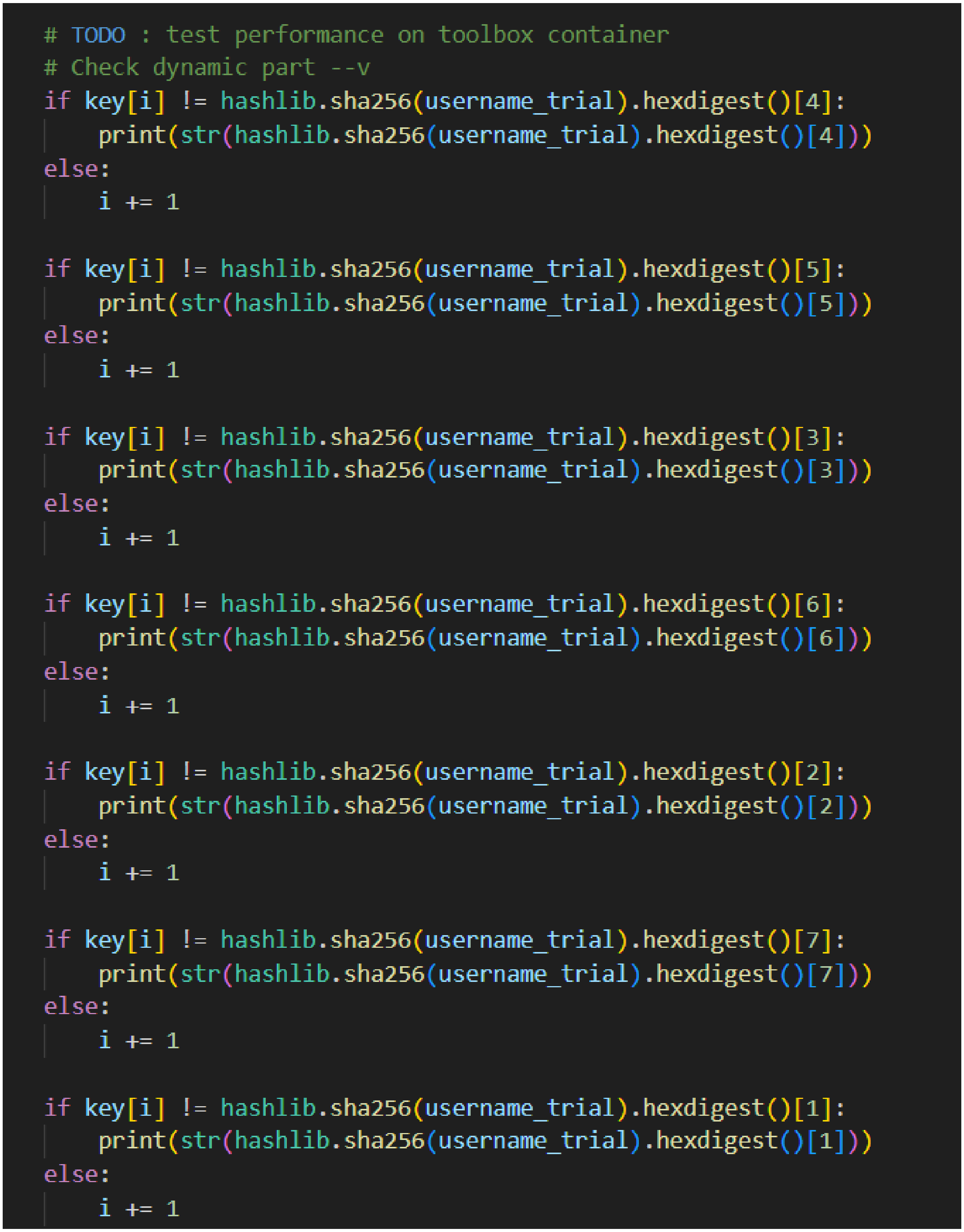
Later on I saw this



```python
# TODO : test performance on toolbox container
# Check dynamic part --v
if key[i] != hashlib.sha256(username_trial).hexdigest()[4]:
    return False
else:
    i += 1

if key[i] != hashlib.sha256(username_trial).hexdigest()[5]:
    return False
else:
    i += 1

if key[i] != hashlib.sha256(username_trial).hexdigest()[3]:
    return False
else:
    i += 1

if key[i] != hashlib.sha256(username_trial).hexdigest()[6]:
    return False
else:
    i += 1

if key[i] != hashlib.sha256(username_trial).hexdigest()[2]:
    return False
else:
    i += 1

if key[i] != hashlib.sha256(username_trial).hexdigest()[7]:
    return False
else:
    i += 1

if key[i] != hashlib.sha256(username_trial).hexdigest()[1]:
    return False
else:
```

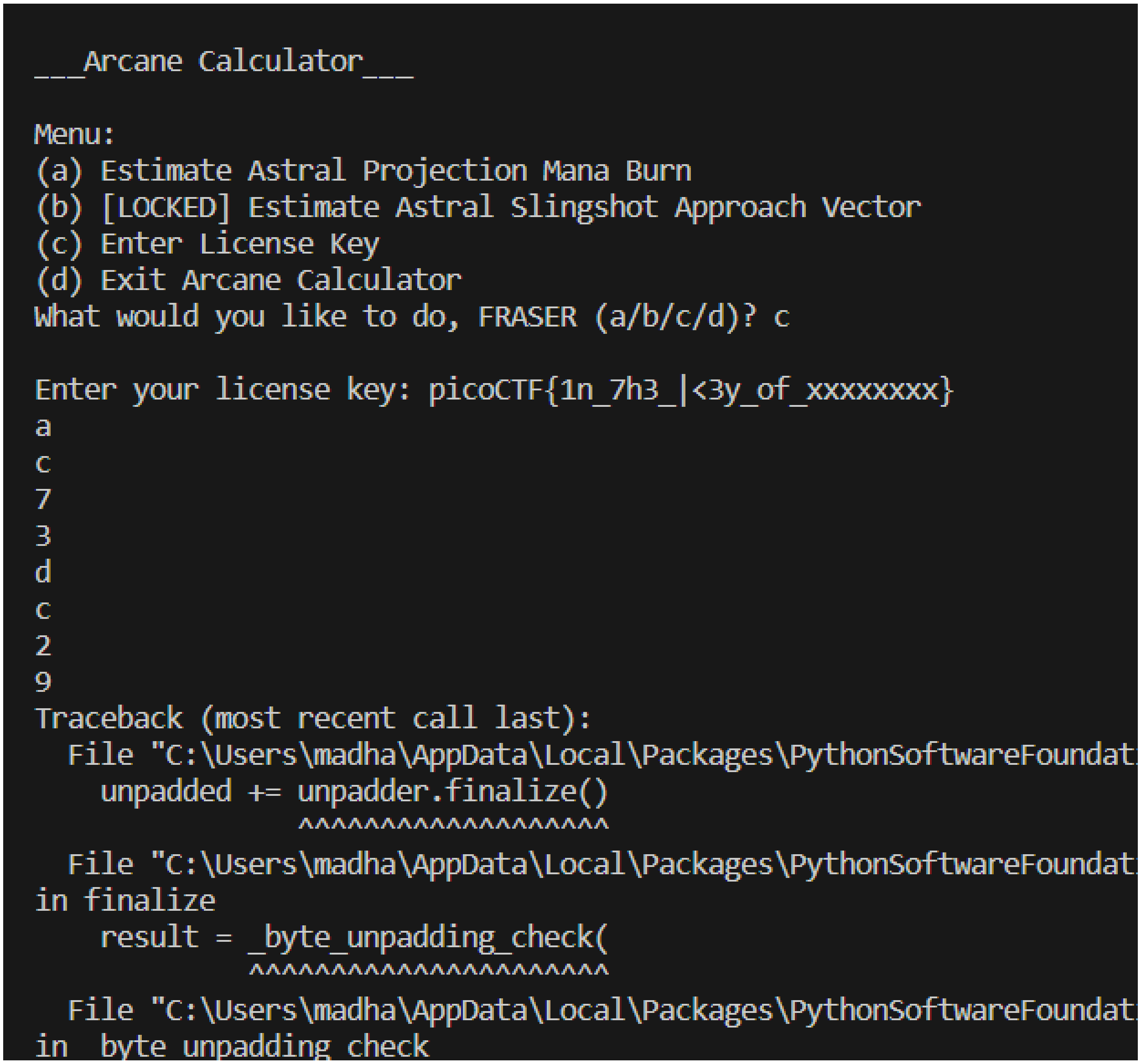
In this part of the code, the program was checking the dynamic part.
It was comparing the elements of entered dynamic key part to some hashlib.sha256whatever
And only giving the go if this hashlib blahblah was entered

So I figured these sha256 hashed things were the correct dynamic key elements, instead of the xxxxsex
So I tweaked the code by making it print the hashed thingies.



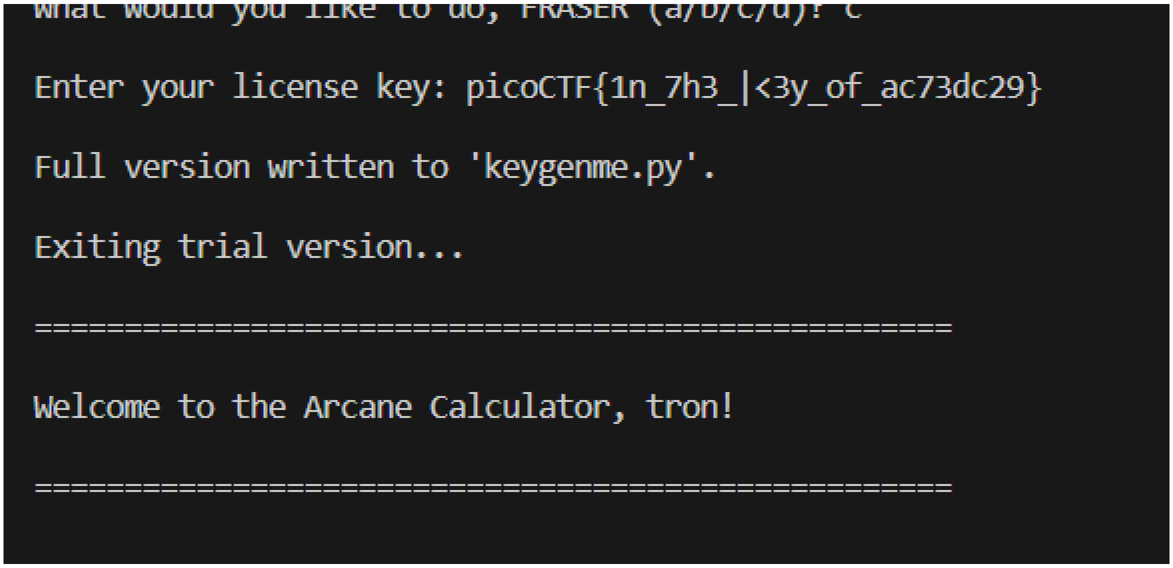
```python
# TODO : test performance on toolbox container
# Check dynamic part --v
if key[i] != hashlib.sha256(username_trial).hexdigest()[4]:
    print(str(hashlib.sha256(username_trial).hexdigest()[4]))
else:
    i += 1

if key[i] != hashlib.sha256(username_trial).hexdigest()[5]:
    print(str(hashlib.sha256(username_trial).hexdigest()[5]))
else:
    i += 1

if key[i] != hashlib.sha256(username_trial).hexdigest()[3]:
    print(str(hashlib.sha256(username_trial).hexdigest()[3]))
else:
    i += 1

if key[i] != hashlib.sha256(username_trial).hexdigest()[6]:
    print(str(hashlib.sha256(username_trial).hexdigest()[6]))
else:
    i += 1

if key[i] != hashlib.sha256(username_trial).hexdigest()[2]:
    print(str(hashlib.sha256(username_trial).hexdigest()[2]))
else:
    i += 1

if key[i] != hashlib.sha256(username_trial).hexdigest()[7]:
    print(str(hashlib.sha256(username_trial).hexdigest()[7]))
else:
    i += 1

if key[i] != hashlib.sha256(username_trial).hexdigest()[1]:
    print(str(hashlib.sha256(username_trial).hexdigest()[1]))
else:
    i += 1
```

Sure enough, when I ran it :



```
___Arcane Calculator___

Menu:
(a) Estimate Astral Projection Mana Burn
(b) [LOCKED] Estimate Astral Slingshot Approach Vector
(c) Enter License Key
(d) Exit Arcane Calculator
What would you like to do, FRASER (a/b/c/d)? c

Enter your license key: picoCTF{1n_7h3_|<3y_of_xxxxxxxx}
a
c
7
3
d
c
2
9
Traceback (most recent call last):
  File "C:\Users\madha\AppData\Local\Packages\PythonSoftwareFoundat
    unpadded += unpadder.finalize()
                ^^^^^^^^^^^^^^^^^^^
  File "C:\Users\madha\AppData\Local\Packages\PythonSoftwareFoundat
in finalize
    result = _byte_unpadding_check(
             ^^^^^^^^^^^^^^^^^^^^^^
  File "C:\Users\madha\AppData\Local\Packages\PythonSoftwareFoundat
in _byte_unpadding_check
```

I then replaced the x's with the printed stuff ac73dc29
And entered this new key into the program aaaaaand :



```
What would you like to do, FRASER (a/b/c/d)? c

Enter your license key: picoCTF{1n_7h3_|<3y_of_ac73dc29}

Full version written to 'keygenme.py'.

Exiting trial version...

===================================================

Welcome to the Arcane Calculator, tron!

===================================================
```

it seems to be the right one.

Put it into pico, and yeay :

| 30 points

picoCTF{1n_7h3_|<3y_of_ac73dc29}