NeverLAN CTF

Programming : Das Prime

Value : 100 pts

Difficulty : Easy

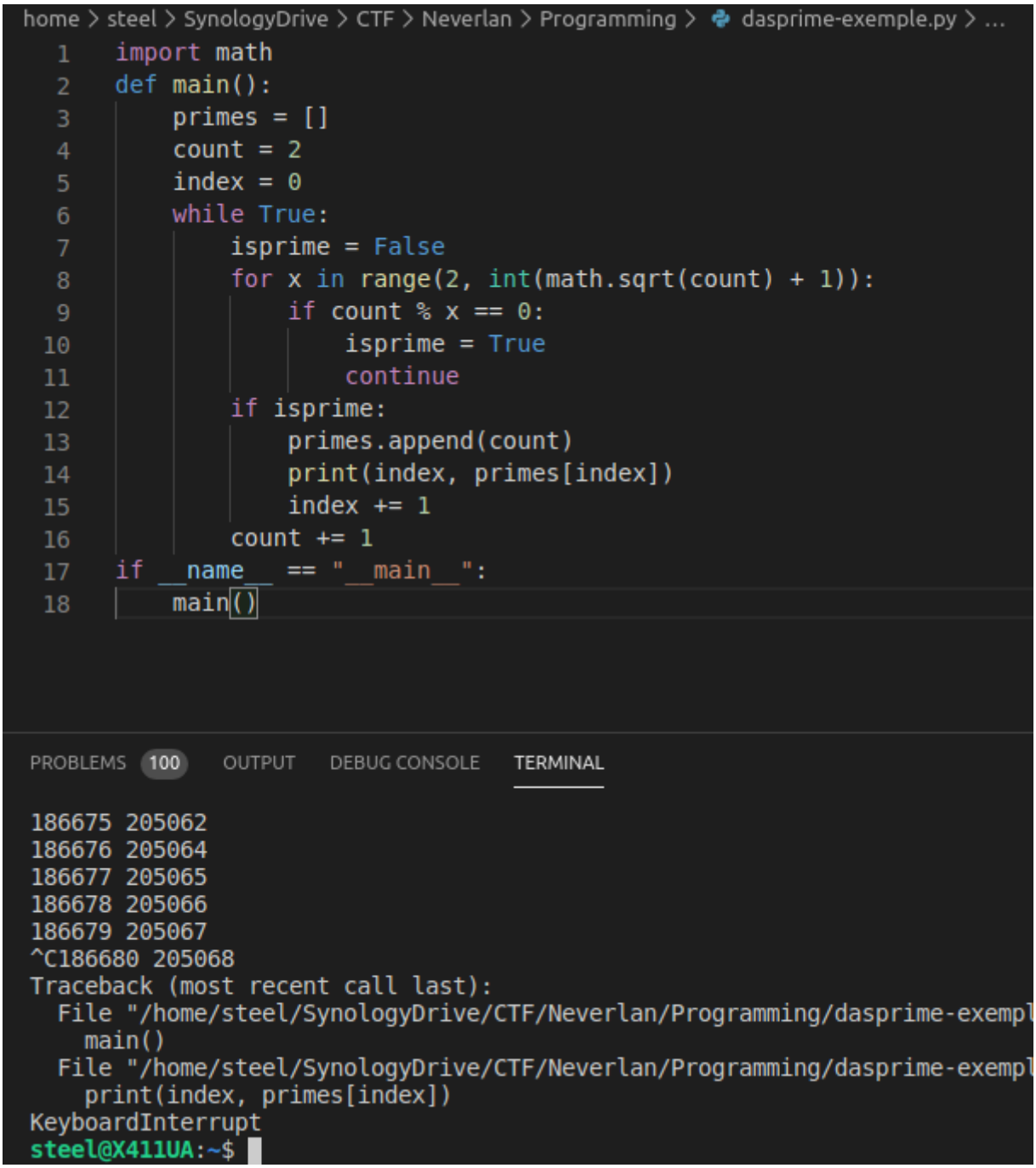
Description : Your flag won't be in the normal flag{flagGoesHere} syntax. Instead you're looking for a prime number.

Attachment : My assignments due and I still don't have the answer! Can you help me fix my Python script... and also give me the answer? I need to make a prime number generator and find the 10,497th prime number. I've already written a python script that kinda works... can you either fix it or write your own and tell me the prime number?

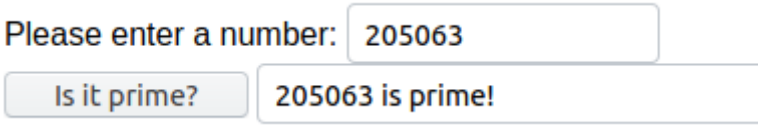
```
import math
def main():
    primes = []
    count = 2
    index = 0
    while True:
        isprime = False
        for x in range(2, int(math.sqrt(count) + 1)):
            if count % x == 0:
                isprime = True
                continue
        if isprime:
            primes.append(count)
            print(index, primes[index])
            index += 1
        count += 1
if __name__ == "__main__":
    main()
```

Solution :

A python script to find prime numbers is given, but it is said that it’s doesn’t really works. Let’s execute it first, to see how it’s broken.



The programs never ends because of the ‘while True’. We’ll modify this later if we want to be able to see the prime number we need.
In the output, the index is the first number on the line, and the prime number is second.
The ‘prime numbers’ in this output are clearly not prime. On the contrary, the not displayed 205063 seems to be one!



This programs output non-prime numbers.
If we check the code used to decide if a number is prime or not, we can see these lines:

```
isprime = False
        for x in range(2, int(math.sqrt(count) + 1)):
            if count % x == 0:
                isprime = True
```

These lines tell to the program that if the number can be divided, then it's prime. They are the lines that we need to change. If the number can be divided, then it's not a prime number. The values of 'isprime' need to be changed.

We add a break at the number 10497, for the flag. The modified program looks like this:

```
import math
def main():
    primes = []
    count = 2
    index = 0
    while True:
        isprime = True
        for x in range(2, int(math.sqrt(count) + 1)):
            if count % x == 0:
                isprime = False
                continue
        if isprime:
            primes.append(count)
            print(index, primes[index])
            index += 1
        count += 1
        if index == 10497:
            break
if __name__ == "__main__":
    main()
```

The output is now :

```
10485 110479
10486 110491
10487 110501
10488 110503
10489 110527
10490 110533
10491 110543
10492 110557
10493 110563
10494 110567
10495 110569
10496 110573
steel@X411UA:~$
```

Because the ‘index +=1’ is after the print, the prime number is displayed as the 10496th one, but it’s in fact the 10497th. The numbers in the output are all prime numbers.

flag{110573}