Just Missed

DiPS CodeJam 23

Prompt

Given a number $n \geq 3$, find the the smallest number k such that the modular residues of k by the first n prime numbers belongs to the set $\{-1, 0, 1\}$.
For example, if $n = 5$, the first 5 primes are $2, 3, 5, 7, 11$. This means that $k \in [10, \infty)$.

- $10 \pmod 7 = 3$, move on to the next number
- $11 \pmod 7 = 4$, move on to the next number
- $12 \pmod 5 = 2$, move on to the next number
- $13 \pmod 5 = 3$, move on to the next number
- $14 \pmod 1 1 = 3$, move on to the next number
- $15 \pmod 1 1 = 4$, move on to the next number
- $16 \pmod 7 = 2$, move on to the next number
- $17 \pmod 5 = 2$, move on to the next number
- $18 \pmod 5 = 3$, move on to the next number
- $19 \pmod 7 = 7$, move on to the next number
- $20 \pmod 1 1 = 9$, move on to the next number
- 21:
  - $21 \pmod 2 = 1$
  - $21 \pmod 3 = 0$
  - $21 \pmod 5 = 1$
  - $21 \pmod 7 = 0$
  - $21 \pmod 1 1 = -1$

Therefore, $k = 21$.

Input Format

The first and only line of input contains n.

Output Format

Your output should consist of one line that contains the number k.

Constraints

$n \geq 3$

Sample Input/Output

Input	Output
5	21

Solution

Refer to the example given.

Sample Program

```
import math

def generateFirstNPrimes(n):
  res=[]
  X = 0
  i = 2
  while True:
    flag=True
    if X==n:
      break

    for j in range(2, i//2+1):
      if i%j==0:
        flag=False
        break
    if flag:
      res.append(i)
      X+=1

    i+=1
  return res


def modWithNegativeOne(dividend, divisor):
  rawMod=dividend%divisor
  return rawMod if rawMod-divisor != -1 else -1

def checkModulos(k,list_of_numbers):
  residues=[modWithNegativeOne(k,i) for i in list_of_numbers]
  return True if set(residues).issubset(set([-1,0,1])) else False


n=int(input())
primes=generateFirstNPrimes(n)
k=primes[-1]-1

while checkModulos(k, primes)==False:
  k+=1
```

```
print(k)
```