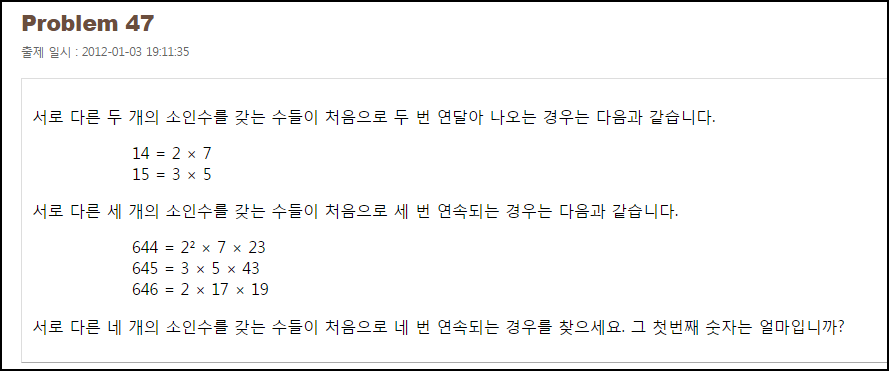
**Exercise**

서로 다른 4개의 소인수를 가지면서, 처음으로 연속되는 경우의 첫번째 숫자를 구하는 문제입니다.



**Solution\_code**

prime 모듈을 이용해서, 간단히 소인수 분해를 리턴 할 수 있었습니다. 리턴되는 소인수 분해 리스트에서 중복되는 소인수 들을 제곱값으로 합쳐주었고, 해당 길이들을 체크하는 방식으로 연속적으로 4개의 소인수를 가질 경우 멈추는 방식입니다.

|  |  |  |
| --- | --- | --- |
| **1**  **2**  **3**  **4**  **5**  **6**  **7**  **8**  **9**  **10**  **11**  **12**  **13**  **14**  **15**  **16**  **17**  **18**  **19**  **20** | **#소인수 분해 , 소수구하기 엄청 간단하다!!!**  **from prime import \***  **#n = input("num : ")**  **#print Prime.factor(n) ##소수면 NONE 리턴**  **#print Prime.factorize(n) ##소인수 분해 결과 리턴**      **def Prime\_duplicate(n):#중복인자들을 하나의 곱으로 표현 후 리턴**  **l = Prime.factorize(n)**  **for i in l:**  **if l.count(i) > 1:**  **l.insert(l.index(i),i\*\*l.count(i))**    **for j in range(l.count(i)):**  **l.remove(i)**  **return l**    **#print Prime\_duplicate(n)**    **Count = 0**  **temp = 0**  **result = set()**  **for i in range(1000,999999):**  **if len(Prime\_duplicate(i)) == 4:#중복합쳐주고, 요소가 4개인가?**  **Count += 1**  **# print Prime\_duplicate(i), i**  **if Count == 4:#연속 4번이 되었는가?**  **print "[+] Find",Prime\_duplicate(i), i, i-1, i-2 ,i-3**  **break**    **else:**  **Count = 0**  ***[Colored by Color Scripter](http://colorscripter.com/info#e)*** | [cs](http://colorscripter.com/info#e) |

**Result**

