

2. 소수구하기

* 이전까지 사용한 방법

• 에라토스테네스의 체

```
1 count = int(input())
2 num = list(map(int, input().split()))
3
4 number = []
5 num_list = [i for i in range(1, 1001)]
6
7 for n in range(1, 1000):
8     if num_list[n] != 0:
9         number.append(num_list[n])
10        for i, v in enumerate(num_list):
11            if num_list[i] % number[-1] == 0:
12                num_list[i] = 0
13
14 answer = 0
15 for i in num:
16     if i in number:
17         answer += 1
18 print(answer)
```

1 ~~2~~ ~~3~~ ~~4~~ ~~5~~ ~~6~~ ~~7~~ ~~8~~ ~~9~~ ~~10~~
11 ~~12~~ ~~13~~ ~~14~~ ~~15~~ ~~16~~ ~~17~~ ~~18~~ ~~19~~ ~~20~~
.
.
.

number = [2, 3]

* 제곱근까지만 구하는 이유

$$\begin{aligned} \text{ex) } 12 &= 1 \times 12 \\ &= 2 \times 6 \\ &= 3 \times 4 \\ &= 2\sqrt{3} \times \sqrt{3} \end{aligned}$$

대칭을 이용

⇒ 제곱근 이전의 숫자에서 약수가 있는지만 확인해도 OK

```
1 import math
2 M, N = map(int, input().split())
3
4 def is_prime(num):
5     if num == 1:
6         return False
7     else:
8         for i in range(2, int(math.sqrt(num))+1):
9             if num % i == 0:
10                 return False
11         return True
12
13 for i in range(M, N+1):
14     if is_prime(i):
15         print(i)
```

$2\sqrt{3} + 1$
 $2, 3 \dots \sqrt{3}$