

# 1차시 문제 풀이

- 정지호

# 1929. 소수 구하기

```
1  def primeNumber(x):
2      if x == 1:
3          return False
4      for i in range(2, int(x**1/2) + 1):
5          if x%i == 0:
6              return False
7      return True
8
9  a, b = map(int, input().split())
10
11  for i in range(a, b+1):
12      if primeNumber(i):
13          print(i)
```

input받아서  
공백 기준으로 split후  
int 형변환 해서  
a, b에 각각 저장

소수란?

**1과 자기 자신을 제외한 약수가  
존재하지 않는 것**

-> 약수가 존재하는지  
확인하려면, **제곱근까지만**  
확인하면 된다. (약수들이  
대칭적으로 짝을 이루기 때문에)

# 1978. 소수 찾기 - 오답

```
1  def primeNumber(x):
2      if x == 1:
3          return False
4      for i in range(2, int(x**1/2) + 1):
5          if x%i == 0:
6              return False
7      return True
8
9  n = int(input())
10 numbers = list(map(int, input().split()))
11 sum = 0
12
13 for i in numbers:
14     if primeNumber(i):
15         sum += 1
16
17 print(sum)
```

문제	결과
 1929	시간 초과

시간복잡도 :  $n^2$

# 1978. 소수 찾기 - 정답

```
1  def prime_list(n):
2      sieve = [True] * (n+1)
3
4      m = int(n ** 0.5)
5      for i in range(2, m + 1):
6          if sieve[i] == True:
7              for j in range(i+i, n+1, i):
8                  sieve[j] = False
9
10     # 소수 목록 산출
11     return [i for i in range(2, n+1) if sieve[i] == True]
12
13
14     a, b = map(int, input().split())
15     primes = prime_list(b)
16
17     for i in primes:
18         if i >= a:
19             print(i)
```

에라토스테네스의 체 사용

일단 b까지의 소수 리스트인  
primes를 만든 뒤,

a보다 크거나 같은 것만 출력

# 4948. 베르트랑 공준

에라토스테네스의 체 사용

```
1  def prime_list(n): # n보다 작거나 같은 소수
2      sieve = [True] * (n+1)
3
4      m = int(n ** 0.5)
5      for i in range(2, m + 1):
6          if sieve[i] == True:
7              for j in range(i+i, n+1, i):
8                  sieve[j] = False
9
10     # 소수 목록 산출
11     return [i for i in range(2, n+1) if sieve[i] == True]
12
13 while True:
14     n = int(input())
15     if n == 0:
16         break
17     sum = 0
18     primes = prime_list(2*n)
19     for i in primes:
20         if i > n:
21             sum += 1
22     print(sum)
```

## 15649. n과 m(1)

```
from itertools import permutations
a, b = map(int, input().split())
nums = list(range(1, a+1)) 범위 안의 수를  
리스트로 만들기

for i in permutations(nums, b):
    for j in range(0, b):
        print(i[j], end=' ')
    print()
```

# 11653. 소인수 분해

```
n = int(input())
primes = prime_list(n)
nums = []

while n > 1:
    for i in primes:
        if n%i == 0:
            nums.append(i)
            n /= i
            continue

nums.sort()
for i in nums:
    print(i)
```

prime\_list : 에라토스테네스의 체 사용

for문을 돌아서 소수 list의  
첫번째 인덱스부터 접근.

n이 요소로 나누어떨어지면

1. 요소를 소인수 리스트에  
추가
2. n을 요소로 나눔
3. 처음부터 반복