

수학 문제

- 정지호

- 2022.09.01

문제집

<https://www.acmicpc.net/workbook/view/8997>

전부 풀이 업로드

너무 오래 걸리는 건 풀지 않기

n까지의 합 (8393번)

합

성공

다국어

☆

한국어 ▾

시간 제한	메모리 제한	제출	정답	맞힌 사람	정답 비율
1 초	128 MB	188661	122143	104757	65.152%

문제

n이 주어졌을 때, 1부터 n까지 합을 구하는 프로그램을 작성하시오.

입력

첫째 줄에 n ($1 \leq n \leq 10,000$)이 주어진다.

출력

1부터 n까지 합을 출력한다.

```
1 sum = 0
2 n = int(input())
3
4 for i in range(1, n+1):
5     sum += i
6
7 print(sum)
```

+ n까지의 곱

```
1 mul = 1
2 n = int(input())
3
4 for i in range(1, n+1):
5     mul *= i
6
7 print(mul)
```

유클리드 호제법으로 최대 공약수 구하기

$106 / 16 = 6, \text{ remainder } 10$
 $16 / 10 = 1, \text{ remainder } 6$
 $10 / 6 = 1, \text{ remainder } 4$
 $6 / 4 = 1, \text{ remainder } 2$
 $4 / 2 = 2, \text{ remainder } 0$
 GCD

(1071, 1029) =

(1029, 42) =

(42, 21) =

(21, 0)

a를 b로 나눈 나머지가 r일 때,

a와 b의 최대 공약수를 (a, b) 라고 하면 다음이 성립한다.

$$(a, b) = (b, r)$$

유클리드 호제법으로 최대 공약수 구하기

$$\begin{array}{l} 106 / 16 = 6, \text{ remainder } 10 \\ 16 / 10 = 1, \text{ remainder } 6 \\ 10 / 6 = 1, \text{ remainder } 4 \\ 6 / 4 = 1, \text{ remainder } 2 \\ 4 / 2 = 2, \text{ remainder } 0 \\ \text{GCD} \end{array}$$

< 1회차 >

$$a = 106$$

$$b = 16$$

$$r = 10$$

< 2회차 >

$$a = 16 = \text{1회차의 } b$$

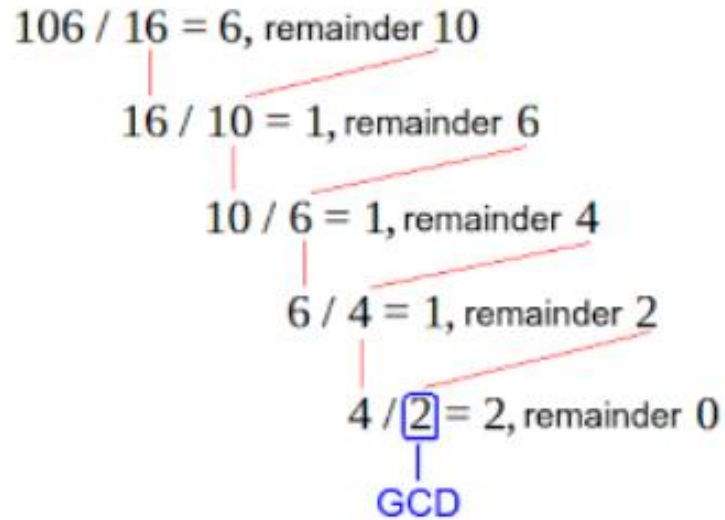
$$b = 10 = \text{1회차의 } r$$

$$r = 6 = \text{1회차의 } a \% b$$

종료조건??

$r = 0$ 일 때의 b 가 답

유클리드 호제법으로 최대 공약수 구하기



코드

```
1 def uclid(a, b):
2     r = a % b
3     if r == 0:           #종료조건
4         return b
5     else:
6         return uclid(b, r)
```

< 1회차 >

< 2회차 >

종료조건??

a = 106

a = 16 = 1회차의 b

r = 0일 때의 b가 답

b = 16

b = 10 = 1회차의 r

r = 10

r = 6 = 1회차의 a%b

소수 문제 : 에라토스테네스의 체

https://ko.wikipedia.org/wiki/%EC%97%90%EB%9D%BC%ED%86%A0%EC%8A%A4%ED%85%8C%EB%84%A4%EC%8A%A4%EC%9D%98_%EC%B2%B4

```
1  def prime_list(n):
2      # 에라토스테네스의 체 초기화: n개 요소에 True 설정(소수로 간주)
3      sieve = [True] * n
4
5      # n의 최대 약수가 sqrt(n) 이하이므로 i=sqrt(n)까지 검사
6      m = int(n ** 0.5)
7      for i in range(2, m + 1):
8          if sieve[i] == True:           # i가 소수인 경우
9              for j in range(i+i, n, i): # i이후 i의 배수들을 False 판정
10                 sieve[j] = False
11
12     # 소수 목록 산출
13     return [i for i in range(2, n) if sieve[i] == True]
```

순열

서로 다른 n개에서 r개를 선택할 때 **순서를 고려하여 중복없이** 뽑을 경우의 수

예) A, B, C, D에서 두 개를 순서를 고려하여 중복없이 뽑을 경우의 수

AB/AC/AD

BA/BC/BD

CA/CB/CD

DA/DB/DC

총 12가지

AB와 BA를 다른 것으로 본다.

모듈 itertools로 구현 가능

```
from itertools import permutations
```

```
arr = ['A', 'B', 'C', 'D']
```

```
permutations(arr, 2)
```

```
: <itertools.permutations at 0x1ffb3413a90>
```

```
for i in permutations(arr, 2):  
    print(i)
```

```
('A', 'B')
```

```
('A', 'C')
```

```
('A', 'D')
```

```
('B', 'A')
```

```
('B', 'C')
```

```
('B', 'D')
```

```
('C', 'A')
```

```
('C', 'B')
```

```
('C', 'D')
```

```
('D', 'A')
```

```
('D', 'B')
```

```
('D', 'C')
```


조합

서로 다른 n개에서 r개를 선택할 때, **순서를 고려하지 않고** 중복없이 뽑는 경우의 수

예를 들어 A B C D의 모든 가능한 조합은 AB/AC/AD

BC/BD

CD

로 총 6가지이다. 순서를 고려하지 않기 때문에 AB, BA는 같은 경우로 본다.

```
from itertools import combinations
```

```
print(combinations(arr,2))
```

```
<itertools.combinations object at 0x000001FFB3444950>
```

```
for i in combinations(arr, 2):  
    print(i)
```

```
('A', 'B')
```

```
('A', 'C')
```

```
('A', 'D')
```

```
('B', 'C')
```

```
('B', 'D')
```

```
('C', 'D')
```