

04 정수론

1. 약수의 합

<문제 설명>

정수 n 을 입력받아 n 의 약수를 모두 더한 값을 리턴하는 함수, `solution`을 완성해주세요.

<제한 조건>

- n 은 0 이상 3000이하인 정수입니다.

<입출력 예>

n	return
12	28
5	6

입출력 예#1

12의 약수는 1, 2, 3, 4, 6, 12입니다. 이를 모두 더하면 28입니다.

입출력 예 #2

5의 약수는 1, 5입니다. 이를 모두 더하면 6입니다.

04 정수론

2. 약수의 개수와 덧셈

<문제 설명>

두 정수 left와 right가 매개변수로 주어집니다. left부터 right까지의 모든 수들 중에서, 약수의 개수가 짝수인 수는 더하고, 약수의 개수가 홀수인 수는 뺀 수를 return 하도록 solution 함수를 완성해주세요.

<제한 조건>

- 1 ≤ left ≤ right ≤ 1,000

<입출력 예>

left	right	result
13	17	43
24	27	52

<입출력 예>

수	약수	약수의 개수
13	1, 13	2
14	1, 2, 7, 14	4
15	1, 3, 5, 15	4
16	1, 2, 4, 8, 16	5
17	1, 17	2

수	약수	약수의 개수
24	1, 2, 3, 4, 6, 8, 12, 24	8
25	1, 5, 25	3
26	1, 2, 13, 26	4
27	1, 3, 9, 27	4

입출력 예#1

13 + 14 + 15 - 16 + 17 = 43을 return

입출력 예 #2

24 - 25 + 26 + 27 = 52를 return

04 정수론

3. 소수 찾기

<문제 설명>

1부터 입력받은 숫자 n 사이에 있는 소수의 개수를 반환하는 함수, solution을 만들어 보세요.

소수는 1과 자기 자신으로만 나누어지는 수를 의미합니다. (1은 소수가 아닙니다.)

<제한 조건>

- n은 2이상 1000000이하의 자연수입니다.

<입출력 예>

n	result
10	4
3	3

입출력 예#1

1부터 10 사이의 소수는 [2,3,5,7] 4개가 존재하므로 4를 반환

입출력 예 #2

1부터 5 사이의 소수는 [2,3,5] 3개가 존재하므로 3를 반환

04 정수론

4. 소수 만들기

<문제 설명>

주어진 숫자 중 3개의 수를 더했을 때 소수가 되는 경우의 개수를 구하려고 합니다. 숫자들이 들어있는 배열 `nums`가 매개변수로 주어질 때, `nums`에 있는 숫자들 중 서로 다른 3개를 골라 더했을 때 소수가 되는 경우의 개수를 `return` 하도록 `solution` 함수를 완성해주세요.

<제한 조건>

- `nums`에 들어있는 숫자의 개수는 3개 이상 50개 이하입니다.
- `nums`의 각 원소는 1 이상 1,000 이하의 자연수이며, 중복된 숫자가 들어있지 않습니다.

<입출력 예>

nums	result
[1,2,3,4]	1
[1,2,7,6,4]	4

입출력 예#1

[1,2,4]를 이용해서 7을 만들 수 있습니다.

입출력 예 #2

[1,2,4]를 이용해서 7을 만들 수 있습니다.

[1,4,6]을 이용해서 11을 만들 수 있습니다.

[2,4,7]을 이용해서 13을 만들 수 있습니다.

[4,6,7]을 이용해서 17을 만들 수 있습니다.

04 정수론

5. 하샤드 수

<문제 설명>

양의 정수 x 가 하샤드 수이라면 x 의 자릿수의 합으로 x 가 나누어져야 합니다. 예를 들어 18의 자릿수 합은 $1+8=9$ 이고, 18은 9로 나누어 떨어지므로 18은 하샤드 수입니다. 자연수 x 를 입력받아 x 가 하샤드 수인지 아닌지 검사하는 함수, `solution`을 완성해주세요.

<제한 조건>

- x 는 1 이상, 10000 이하인 정수입니다.

<입출력 예>

arr	return
10	true
12	true
11	false
13	false

입출력 예#1

10의 모든 자릿수의 합은 1입니다. 10은 1로 나누어 떨어지므로 10은 하샤드 수입니다.

입출력 예 #2

12의 모든 자릿수의 합은 3입니다. 12는 3으로 나누어 떨어지므로 12는 하샤드 수입니다.

입출력 예#3

11의 모든 자릿수의 합은 2입니다. 11은 2로 나누어 떨어지지 않으므로 11는 하샤드 수가 아닙니다.

입출력 예 #4

13의 모든 자릿수의 합은 4입니다. 13은 4로 나누어 떨어지지 않으므로 13은 하샤드 수가 아닙니다.

04 정수론

6. 콜라츠 추측

<문제 설명>

1937년 Collatz란 사람에 의해 제기된 이 추측은, 주어진 수가 1이 될 때까지 다음 작업을 반복하면, 모든 수를 1로 만들 수 있다는 추측이며, 작업은 다음과 같습니다.

- 1-1. 입력된 수가 짝수라면 2로 나눕니다.
- 1-2. 입력된 수가 홀수라면 3을 곱하고 1을 더합니다.
- 2. 결과로 나온 수에 같은 작업을 1이 될 때까지 반복합니다.

예를 들어, 주어진 수가 6이라면 $6 \rightarrow 3 \rightarrow 10 \rightarrow 5 \rightarrow 16 \rightarrow 8 \rightarrow 4 \rightarrow 2 \rightarrow 1$ 이 되어 총 8번 만에 1이 됩니다. 위 작업을 몇 번이나 반복해야 하는지 반환하는 함수, solution을 완성해 주세요. 단, 주어진 수가 1인 경우에는 0을, 작업을 500번 반복할 때까지 1이 되지 않는다면 -1을 반환해 주세요

<제한 조건>

- 입력된 수, num은 1 이상 8,000,000 미만인 정수입니다.

<입출력 예>

n	result
6	8
16	4
626331	-1

입출력 예#2

$16 \rightarrow 8 \rightarrow 4 \rightarrow 2 \rightarrow 1$ 이 되어 총 4번 만에 1이 됩니다.

입출력 예 #3

626331은 500번을 시도해도 1이 되지 못하므로 -1을 리턴해야 합니다.

04 정수론

7. 최대공약수와 최소공배수

<문제 설명>

두 수를 입력받아 두 수의 최대공약수와 최소공배수를 반환하는 함수, `solution`을 완성해 보세요. 배열의 맨 앞에 최대공약수, 그다음 최소공배수를 넣어 반환하면 됩니다. 예를 들어 두 수 3, 12의 최대공약수는 3, 최소공배수는 12이므로 `solution(3, 12)`는 `[3, 12]`를 반환해야 합니다.

<제한 조건>

- 두 수는 1이상 1000000이하의 자연수입니다.

<입출력 예>

n	m	return
3	12	[3, 12]
2	5	[1, 10]

입출력 예#2

자연수 2와 5의 최대공약수는 1, 최소공배수는 10이므로 `[1, 10]`을 리턴해야 합니다.