1. N개의 최소공배수

def solution(arr):

    arrmax = 1

    allpass = 0

    answer = 0

    for maxnum in arr:

        arrmax \*= maxnum

    for i in range(max(arr), arrmax+1):

        for j in range(0,len(arr)):

            if i % arr[j] == 0:

                allpass += 1

        if allpass == len(arr):

            answer = i

            break

        allpass = 0

    return answer

arr = [1,2,3]

print(solution(arr))

<https://programmers.co.kr/learn/courses/30/lessons/12953>

1. 튜플

def solution(s):

    answer = []

    result = []

    result1 = []

    num = ""

    for i in range(len(s)-1):

        if s[i].isdigit() and s[i+1].isdigit():

            num += s[i]

        elif s[i].isdigit() and not s[i+1].isdigit():

            num += s[i]

            result1.append(num)

            num = ""

        if s[i] == "}":

            result.append(result1)

            result1 = []

    result.sort(key=lambda x:len(x))

    for i in range(len(result)):

        for j in range(len(result[i])):

            if int(result[i][j]) not in answer:

                answer.append(int(result[i][j]))

    return answer

s = "{{4,2,3},{3},{2,3,4,1},{2,3}}"

print(solution(s))

<https://programmers.co.kr/learn/courses/30/lessons/64065>

1. 행렬의 곱셈

def solution(arr1, arr2):

    x = len(arr2[0])

    y = len(arr1)

    answer = []

    for i in range(y):

        x\_list = []

        for j in range(x):

            number = 0

            for k in range(len(arr1[0])):

                nf1 = arr1[i][k]

                nf2 = arr2[k][j]

                number += nf1 \* nf2

            x\_list.append(number)

        answer.append(x\_list)

    return answer

arr1 = [[1, 4], [3, 2], [4, 1]]

arr2 = [[3, 3], [3, 3]]

print(solution(arr1,arr2))

https://programmers.co.kr/learn/courses/30/lessons/12949