1. 신고 결과 받기

def solution(id\_list, report, k):

  answer = [0] \* len(id\_list)

  report = set(report)

  dict\_user\_id = {}

  dict\_user\_report = {}

  for i in id\_list:

    dict\_user\_id[i] = 0

    dict\_user\_report[i] = ""

  for j in report:

    dict\_user\_id[j.split()[1]] += 1

    dict\_user\_report[j.split()[0]] += j.split()[1] + " "

  for a in range(0,len(id\_list)):

    for l in dict\_user\_report[id\_list[a]].split():

      if dict\_user\_id[l] >= k:

      answer[a] += 1

  return answer

k=2

list1 = ["muzi", "frodo", "apeach", "neo"]

list2 = ["muzi frodo","apeach frodo","frodo neo","muzi neo","apeach muzi"]

list3 = ["con", "ryan"]

list4 = ["ryan con", "ryan con", "ryan con", "ryan con"]

print(solution(list1,list2,k))

print(solution(list3, list4, k))

1. 로또의 최고 순위와 최저 순위

def solution(lottos, win\_nums):

    answer = [0,0]

    result = [6,6,5,4,3,2,1]

    zeroCount = 0

    match = 0

    for i in lottos:

      if i == 0:

        zeroCount += 1

      if i in win\_nums:

        match += 1

    else:

      answer[0] = result[zeroCount+ match]

      answer[1] = result[match]

    return answer

list1 = [45, 4, 35, 20, 3, 9]

list2 = [20, 9, 3, 45, 4, 35]

print(solution(list1, list2))

https://programmers.co.kr/learn/courses/30/lessons/77484

1. 신규 아이디 추천

def solution(new\_id):

    answer = ''

    new\_id = new\_id.lower() #소문자로

    for c in new\_id:

      if c.isalpha() or c.isdigit() or c in "-\_.":

        answer += c

    while ".." in answer:

      answer = answer.replace("..",".")

    if answer[0] == ".":

      if len(answer) >= 2:

        answer = answer[1:]

      else:

        answer="."

    if answer[-1] == ".":

      answer = answer[:-1]

    if answer == "":

      answer = "a"

    if len(answer) >= 16:

      answer = answer[:15]

      if answer[-1] == ".":

        answer = answer[:-1]

    while len(answer) < 3:

      answer += answer[-1]

    return answer

id = "...!@BaT#\*..y.abcdefghijklm"

print(solution(id))

<https://programmers.co.kr/learn/courses/30/lessons/72410>

1. 숫자 문자열과 영단어

def solution(s):

    i=0

    now = ""

    answer = ""

    numlist = {

        "zero":"0",

        "one":"1",

        "two":"2",

        "three":"3",

        "four":"4",

        "five":"5",

        "six":"6",

        "seven":"7",

        "eight":"8",

        "nine":"9"

    }

    while i < len(s):

        if s[i].isdigit():

            answer += s[i]

        else:

            now += s[i]

            if now in numlist:

                answer += numlist[now]

                now = ""

        i += 1

    return int(answer)

randnum = "123"

print(solution(randnum))

[https://programmers.co.kr/learn/courses/30/lessons/81301#](https://programmers.co.kr/learn/courses/30/lessons/81301)

1. 키패드 누르기

def solution(numbers, hand):

    answer = ''

    dic = {1: [0, 0], 2: [0, 1], 3: [0, 2],

        4: [1, 0], 5: [1, 1], 6: [1, 2],

        7: [2, 0], 8: [2, 1], 9: [2, 2],

        '\*':[3, 0], 0: [3, 1], '#': [3, 2]}

    left\_hand = dic['\*']

    right\_hand = dic['#']

    for i in numbers:

        now = dic[i]

        if i in [1,4,7]:

            answer += "L"

            left\_hand = now

        elif i in [3,6,9]:

            answer += "R"

            right\_hand = now

        else:

            left\_d = 0

            right\_d = 0

            # 좌표 거리 계산해주기

            for a, b, c in zip(left\_hand, right\_hand, now):

                left\_d += abs(a-c)

                right\_d += abs(b-c)

            # 왼손이 더 가까운 경우

            if left\_d < right\_d:

                answer += 'L'

                left\_hand = now

            # 오른손이 더 가까운 경우

            elif left\_d > right\_d:

                answer += 'R'

                right\_hand = now

            # 두 거리가 같은 경우

            else:

                # 왼손잡이 경우

                if hand == 'left':

                    answer += 'L'

                    left\_hand = now

                # 오른손잡이 경우

                else:

                    answer += 'R'

                    right\_hand = now

    return answer

case1 = [1, 3, 4, 5, 8, 2, 1, 4, 5, 9, 5]

hand = "right"

print(solution(case1, hand))

<https://programmers.co.kr/learn/courses/30/lessons/67256>

1. 크레인 인형뽑기 게임

def solution(board, moves):

    answer = 0

    result = []

    for a in moves:

        for i in range(0,len(board)):

            if board[i][a-1] != 0:

                result.append(board[i][a-1])

                board[i][a-1] = 0

                if len(result) > 1:

                    if result[len(result)-2] == result[len(result)-1]:

                        result.pop()

                        result.pop()

                        answer += 2

                break

    return answer

board = [

    [0,0,0,0,0],

    [0,0,1,0,3],

    [0,2,5,0,1],

    [4,2,4,4,2],

    [3,5,1,3,1]

]

moves = [1,5,3,5,1,2,1,4]

print(solution(board, moves))

-다른 사람이 풀은 답 이랑 비슷해서 뿌듯하다.

<https://programmers.co.kr/learn/courses/30/lessons/64061>

1. 없는 숫자 더하기

def solution(numbers):

    num = [1,2,3,4,5,6,7,8,9]

    answer = 0

    for i in num:

        if not i in numbers:

            answer += i

    return answer

case1 = [1,2,3,4,6,7,8,0]

case2 = [5,8,4,0,6,7,9]

print(solution(case1))

print(solution(case2))

<https://programmers.co.kr/learn/courses/30/lessons/86051>

1. 음양 더하기

def solution(absolutes, signs):

    answer = 0

    for i in range(0, len(absolutes)):

        if signs[i] == True:

            answer += absolutes[i]

        if signs[i] == False:

            answer -= absolutes[i]

    return answer

case1 = [4,7,12]

case2 = [1,2,3]

signs1 = [True,False,True]

signs2 = [False,False,True]

print(solution(case1, signs1))

print(solution(case2, signs2))

<https://programmers.co.kr/learn/courses/30/lessons/76501>

1. 내적

def solution(a, b):

    answer = 0

    for i in range(len(a)):

        answer += a[i] \* b[i]

    return answer

caseA = [1,2,3,4]

caseA2 = [-1,0,1]

caseB = [-3,-1,0,2]

caseB2 = [1,0,-1]

print(solution(caseA, caseB))

print(solution(caseA2, caseB2))

<https://programmers.co.kr/learn/courses/30/lessons/70128>

1. 소수 만들기

def check(a, b, c):

    total = a + b + c

    for i in range(2, total):

        if total % i == 0 : return False

    return True

def solution(nums):

    answer = 0

    for i in range(0, len(nums) - 2):

        for j in range(i+1, len(nums) - 1):

            for k in range(j+1, len(nums)):

                if check(nums[i], nums[j], nums[k]): answer += 1

    return answer

case1 = [1,2,3,4]

case2 = [1,2,7,6,4]

print(solution(case1))

print(solution(case2))

<https://programmers.co.kr/learn/courses/30/lessons/12977>

1. 예산

def solution(d, budget):

    answer = 0

    d.sort()

    for i in range(len(d)):

        if d[i] <= budget:

            answer += 1

            budget -= d[i]

        else:

            break

    return answer

d = [1,3,2,5,4]

budget = 9

print(solution(d,budget))

<https://programmers.co.kr/learn/courses/30/lessons/12982>

1. 약수의 개수와 덧셈

def solution(left, right):

    result = []

    answer = 0

    for i in range(left, right+1):

        result.append(0)

        for j in range(1,i+1):

            if(i % j == 0):

                result[len(result)-1] += 1

        if result[len(result)-1] % 2 == 0:

            answer += i

        else:

            answer -= i

    return answer

left1 = 13

left2 = 24

right1 = 17

right2 = 27

print(solution(left1,right1))

print(solution(left2,right2))

<https://programmers.co.kr/learn/courses/30/lessons/77884>

1. K번째수

def solution(array, commands):

    arr = []

    answer = []

    for i in commands:

        arr = array[i[0]-1:i[1]]

        arr.sort()

        answer.append(arr[i[2]-1])

    return answer

case1 = [1, 5, 2, 6, 3, 7, 4]

commands1 = [[2, 5, 3], [4, 4, 1], [1, 7, 3]]

print(solution(case1, commands1))

<https://programmers.co.kr/learn/courses/30/lessons/42748>

1. 두 개 뽑아서 더하기

def solution(numbers):

    result = []

    answer = []

    for i in range(0, len(numbers)-1):

        for j in range(i+1,len(numbers)):

            result.append(numbers[i] + numbers[j])

    answer = list(set(result))

    answer.sort()

    return answer

numbers1 = [2,1,3,4,1]

numbers2 = [5,0,2,7]

print(solution(numbers1))

print(solution(numbers2))

<https://programmers.co.kr/learn/courses/30/lessons/68644>

1. 3진법 뒤집기

def solution(n):

    base = ""

    answer = 0

    while n>0:

        n,mod = divmod(n,3)

        base += str(mod)

    answer = int(base,3)

    return answer

n = 45

print(solution(n))

<https://programmers.co.kr/learn/courses/30/lessons/68935>

1. 실패율

def solution(N, stages):

    answer = []

    all\_num = len(stages)

    num = {}

    for i in range(1, N+1):

        cnt = 0

        for step in stages:

            if step == i:

                cnt += 1

        if cnt == 0:

            num[i] = 0

        else:

            num[i] = (cnt/all\_num)

        all\_num = all\_num - cnt

    answer = sorted(num, key=lambda x:num[x], reverse = True)

    return answer

N1 = 5

N2 = 4

stages1 = [2, 1, 2, 6, 2, 4, 3, 3]

stages2 = [4,4,4,4,4]

print(solution(N1,stages1))

print(solution(N2,stages2))

시간초과 때문에 정말 어려웠다 ㅋㅋㅋ sorted를 사용하는 방법도 잘 연습해둬야 겠다.

https://programmers.co.kr/learn/courses/30/lessons/42889