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>

1. 2016년

def solution(a, b):

    mounth = {1:31,2:29,3:31,4:30,5:31,6:30,7:31,8:31,9:30,10:31,11:30,12:31}

    total\_day = 0

    first\_Day = 0

    answer = ''

    day = ["FRI","SAT",'SUN',"MON","TUE","WED","THU"]

    for i in range(0,a):

        if i != 0:

            total\_day += mounth[i]

    total\_day += b-1

    f,c = divmod(total\_day, 7)

    print(total\_day)

    print(f)

    print(c)

    if first\_Day + c > 6:

        answer += day[(first\_Day+c)-5]

    else:

        answer += day[first\_Day+c]

    return answer

print(solution(5,24))

<https://programmers.co.kr/learn/courses/30/lessons/12901>

1. 최소직사각형

def solution(sizes):

    result = [0,0]

    for i in range(0,len(sizes)):

        if sizes[i][0] < sizes[i][1]:

            sizes[i] = [sizes[i][1],sizes[i][0]]

        if result[0] < sizes[i][0]:

            result[0] = sizes[i][0]

        if result[1] < sizes[i][1]:

            result[1] = sizes[i][1]

    answer = result[0] \* result[1]

    return answer

sizes1 = [[60, 50], [30, 70], [60, 30], [80, 40]]

sizes2 = [[10, 7], [12, 3], [8, 15], [14, 7], [5, 15]]

sizes3 = [[14, 4], [19, 6], [6, 16], [18, 7], [7, 11]]

print(solution(sizes1))

print(solution(sizes2))

print(solution(sizes3))

<https://programmers.co.kr/learn/courses/30/lessons/86491>

1. 나머지가 1이 되는 수 찾기

def solution(n):

    for i in range(1,n):

        if n%i==1:

            return i

n1 = 10

print(solution(n1))

<https://programmers.co.kr/learn/courses/30/lessons/87389>

1. 부족한 금액 계산하기

def solution(price, money, count):

    answer = 0

    result = 0

    for i in range(1,count+1):

        result += price \* i

    if money < result:

        answer = result - money

    return answer

price1 = 3

money1 = 20

count1 = 4

print(solution(price1,money1,count1))

<https://programmers.co.kr/learn/courses/30/lessons/82612>

1. 비밀지도

def solution(n, arr1, arr2):

    answer = []

    for num1, num2 in zip(arr1,arr2):

        tmp = str(bin(num1 | num2))[2:]

        if len(tmp) < n:

            tmp = '0'\*(n-len(tmp)) + tmp

        tmp = tmp.replace('1','#')

        tmp = tmp.replace('0',' ')

        answer.append(tmp)

    return answer

n1 = 6

arr1\_1 = [46, 33, 33 ,22, 31, 50]

arr2\_2 = [27 ,56, 19, 14, 14, 10]

print(solution(n1,arr1\_1,arr2\_2))

<https://programmers.co.kr/learn/courses/30/lessons/17681>

1. 가운데 글자 가져오기

def solution(s):

    answer = ''

    if len(s) % 2 == 0:

        answer = s[(len(s)//2)-1] + s[len(s)//2]

    else:

        answer = s[len(s)//2]

    return answer

s1 = "abcde"

s2 = "qwer"

solution(s1)

solution(s2)

<https://programmers.co.kr/learn/courses/30/lessons/12903>

1. 나누어 떨어지는 숫자 배열

def solution(arr, divisor):

    answer = []

    for i in arr:

        if i % divisor == 0:

            answer.append(i)

    answer.sort()

    if len(answer) == 0:

        answer.append(-1)

    return answer

arr1 = [5, 9, 7, 10]

arr2 = [2, 36, 1, 3]

arr3 = [3,2,6]

divisor1 = 5

divisor2 = 1

divisor3 = 10

print(solution(arr1,divisor1))

print(solution(arr2,divisor2))

print(solution(arr3,divisor3))

<https://programmers.co.kr/learn/courses/30/lessons/12910>

1. [1차] 다트 게임

def solution(dartResult):

    answer = 0

    result = []

    num = ""

    area = {"S":1,"D":2,"T":3}

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

        if dartResult[i].isdigit():

            num += dartResult[i]

        if dartResult[i].isalpha():

            num = int(num)\*\*int(area[dartResult[i]])

            result.append(num)

            num = ""

        if dartResult[i].isalnum() == False:

            if dartResult[i] == "\*" and len(result) != 1:

                result[len(result)-1] \*= 2

                result[len(result)-2] \*= 2

            elif dartResult[i] == "\*" and len(result) == 1:

                result[len(result)-1] \*= 2

        if dartResult[i].isalnum() == False:

            if dartResult[i] == "#":

                result[len(result)-1] \*= -1

    for re in result:

        answer += re

    return answer

dartResult1 = "1T2D3D#"

print(solution(dartResult1))

<https://programmers.co.kr/learn/courses/30/lessons/17682>

1. 두 정수 사이의 합

def solution(a, b):

    answer = 0

    temp = 0

    if a > b:

        temp = a

        a = b

        b = temp

    for i in range(a,b+1):

        answer += i

    return answer

a = 5

b = 3

print(solution(a,b))

<https://programmers.co.kr/learn/courses/30/lessons/12912>

1. 같은 숫자는 싫어

def solution(arr):

    tmp = []

    answer = []

    result = []

    for i in range(0, len(arr)):

        if i == 0:

            result.append(arr[i])

        else:

            if arr[i-1] == arr[i]:

                result.append(arr[i])

            else:

                tmp.append(list(set(result)))

                result =[]

                result.append(arr[i])

    if len(result) != 0:

        tmp.append(list(set(result)))

    for j in tmp:

        answer.append(j[0])

    return answer

arr1 = [1,1,3,3,0,1,1]

print(solution(arr1))

<https://programmers.co.kr/learn/courses/30/lessons/12906>

1. 문자열 내 마음대로 정렬하기

def solution(strings, n):

    return sorted(sorted(strings), key= lambda x:x[n])

strings1 = ["sun", "bed", "car"]

strings2 = ["abce", "abcd", "cdx"]

n1 = 1

n2 = 2

print(solution(strings1,n1))

print(solution(strings2,n2))

<https://programmers.co.kr/learn/courses/30/lessons/12915>

1. 문자열 내림차순으로 배치하기

def solution(s):

    result = []

    a = ""

    for i in s:

        result.append(i)

    result = sorted(result, reverse=True)

    print(result)

    for j in result:

        a += j

    return a

s1 = "Zbcdefg"

print(solution(s1))

<https://programmers.co.kr/learn/courses/30/lessons/12917>

1. 문자열 다루기 기본

def solution(s):

    answer = True

    length = len(s)

    if length == 4 or length == 6:

        for j in s:

            if j.isdigit() != True:

                answer = False

                return answer

    else:

        answer = False

    return answer

s1 = "1234"

print(solution(s1))

<https://programmers.co.kr/learn/courses/30/lessons/12918>

1. 문자열을 정수로 바꾸기

def solution(s):

    result = ""

    a = ""

    answer = 0

    for i in s:

        if i.isdigit() == True:

            result += i

        else:

            a += i

    answer = int(result)

    if a == "-":

        answer \*= -1

    return answer

s1 = "+1234"

print(solution(s1))

<https://programmers.co.kr/learn/courses/30/lessons/12925>

1. 행렬의 덧셈

def solution(arr1, arr2):

    answer = []

    result = []

    for i, j in zip(arr1,arr2):

        for i\_2, j\_2 in zip(i,j):

            result.append(i\_2 + j\_2)

        answer.append(result)

        result = []

    return answer

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

arr2 = [[3,4],[5,6]]

print(solution(arr1,arr2))

<https://programmers.co.kr/learn/courses/30/lessons/12950>

1. 핸드폰 번호 가리기

def solution(phone\_number):

    answer = len(phone\_number)

    last = phone\_number[-4:]

    return "\*"\*(answer-4) + last

phone\_number1 = "01033334444"

print(solution(phone\_number1))

<https://programmers.co.kr/learn/courses/30/lessons/12948>

1. 짝수와 홀수

def solution(num):

    if num % 2 == 0:

        return "Even"

    else:

        return "Odd"

num1 = 3

num2 = 4

print(solution(num1))

print(solution(num2))

<https://programmers.co.kr/learn/courses/30/lessons/12937>

1. 문자열 내 p와 y의 개수

def solution(s):

    s = s.lower()

    count = [0,0]

    answer = True

    for i in s:

        if i in "PpYy":

            if i == 'p':

                count[0] +=1

            elif i == 'y':

                count[1] +=1

        else:

            answer = True

    if count[0] == count[1] and len(count) != 0:

        answer = True

    else:

        answer = False

    return answer

s1 = ""

print(solution(s1))

<https://programmers.co.kr/learn/courses/30/lessons/12916>

1. 정수 내림차순으로 배치하기

def solution(n):

    result = []

    answer = ""

    for i in str(n):

        result.append(i)

    result = sorted(result, reverse=True)

    for j in result:

        answer += j

    return int(answer)

n = 118372

print(solution(n))

<https://programmers.co.kr/learn/courses/30/lessons/12933>

1. 평균 구하기

def solution(arr):

    answer = 0

    for i in arr:

        answer += i

    answer = answer / len(arr)

    return answer

arr = [1,2,3,4]

print(solution(arr))

<https://programmers.co.kr/learn/courses/30/lessons/12944>

1. 이상한 문자 만들기

def solution(s):

    answer = ''

    new\_list = s.split(" ")

    for i in new\_list:

        for j in range(0, len(i)):

            if j % 2 == 0:

                answer += i[j].upper()

            else:

                answer += i[j].lower()

        answer+= " "

    return answer[0:-1]

s = "try hello world"

print(solution(s))

@ 홀수 기준으로 소문자로 만들랬는데 안해서 틀렸었다.

<https://programmers.co.kr/learn/courses/30/lessons/12930#>

1. 직사각형 별찍기

a, b = map(int, input().strip().split(' '))

for i in range(b):

    for j in range(a):

        print('\*', end='')

    print()

:: 이건 뭔지 ::

<https://programmers.co.kr/learn/courses/30/lessons/12969>

1. 자연수 뒤집어 배열로 만들기

def solution(n):

    answer = []

    for i in str(n):

        answer.append(int(i))

    answer.reverse()

    return answer

n = 12345

print(solution(n))

<https://programmers.co.kr/learn/courses/30/lessons/12932>

1. 자릿수 더하기

def solution(n):

    answer = 0

    for i in str(n):

        answer += int(i)

    return answer

n = 123

print(solution(n))

<https://programmers.co.kr/learn/courses/30/lessons/12931>

1. 수박수박수박수박수박수?

def solution(n):

    answer = ''

    for i in range(n):

        if i % 2 == 0:

            answer += "수"

        else:

            answer += "박"

    return answer

n = 3

print(solution(n))

<https://programmers.co.kr/learn/courses/30/lessons/12922>

1. 서울에서 김서방 찾기

def solution(seoul):

    answer = ''

    for i in range(len(seoul)):

        if seoul[i] in "Kim":

            answer = "김서방은 {0}에 있다".format(i)

    return answer

seoul = ["Jane", "Kim"]

print(solution(seoul))

<https://programmers.co.kr/learn/courses/30/lessons/12919>

1. 약수의 합

def solution(n):

    answer = 0

    for i in range(0, n):

        if n % (i+1) == 0:

            answer += i+1

    return answer

n = 12

print(solution(n))

<https://programmers.co.kr/learn/courses/30/lessons/12928>

1. x만큼 간격이 있는 n개의 숫자

def solution(x, n):

    answer = [0] \* n

    for i in range(n):

        if i == 0:

            answer[i] = x

        else:

            answer[i] = answer[i-1] + x

    return answer

x = -4

n = 2

print(solution(x,n))

<https://programmers.co.kr/learn/courses/30/lessons/12954>

1. 제일 작은 수 제거하기

def solution(arr):

    temp = arr[0]

    answer = []

    for i in range(len(arr)):

        if temp > arr[i]:

            temp = arr[i]

    for j in arr:

        if j != temp:

            answer.append(j)

    if len(answer) == 0:

        answer.append(-1)

    return answer

arr = [10]

print(solution(arr))

<https://programmers.co.kr/learn/courses/30/lessons/12935>

1. 정수 제곱근 판별

from math import sqrt

def solution(n):

    answer = 0

    num = int(sqrt(n))

    if num\*\*2 == n:

        answer = (num+1)\*\*2

    else:

        answer = -1

    return answer

n = 81

print(solution(n))

<https://programmers.co.kr/learn/courses/30/lessons/12934>

1. 소수 찾기

from math import sqrt

def solution(n):

    answer = []

    l = False

    for i in range(2,n+1):

        sq = int(sqrt(i))

        if sq == 1:

            answer.append(i)

        else:

            for j in range(2,sq+1):

                if i % j == 0:

                    l = False

                    break

                else:

                    l = True

            if l == True:

                answer.append(i)

                l = False

    return len(answer)

n = 10

print(solution(n))

## 정말 어려웠다. 소수는 에라토스테네스의 체 알고리즘을 이용해서 푸는게 좋다.

https://programmers.co.kr/learn/courses/30/lessons/12921