checkio - 05\_Between\_Markers

//보경

def between\_markers(text: str, begin: str, end: str) -> str:  
   if text.find(begin)!= -1:  
       start\_index = text.find(begin)+len(begin) # marker의 길이만큼 index를 뒤로 밀어줘야 marker 다음의 문자열을 알 수 있음  
   else:  
       start\_index = 0   if text.find(end)!= -1:  
       end\_index = text.find(end)  
   else:  
       end\_index = len(text)   return text[start\_index:end\_index]

// 민주

def between\_markers(s, marker\_i, marker\_f):   if marker\_i not in s:  
       return s.split(marker\_f)[0]  
   else:  
       return s.split(marker\_i)[1].split(marker\_f)[0]

checkio - 06\_Best\_Stock

//보경

def best\_stock(data):  
   # sorted\_stock = sorted(data, key=lambda x:x[1], reverse=True)  
   # return sorted\_stock[0]   highest\_price = 0  
   for i in data:  
       if data[i] > highest\_price:  
           highest\_price = data[i]  
           highest\_stock = i   return highest\_stock

// 민주

def best\_stock(info):  
   return list(info.keys())[list(info.values()).index(max(info.values()))]

checkio - 07\_Popular\_Words

//보경

def popular\_words(text: str, words: list) -> dict:  
   result = {} #빈 dictionary 생성  
   lower\_text = text.lower()  
   edit\_text = lower\_text.replace('\n', ' ')  
   split\_text = edit\_text.split(' ')   for i in words:  
       result[i] = split\_text.count(i)   # key : i, value : split\_text.count(i)  
   return result

//민주

def popular\_words(text, word):  
   answer = {}   for w in word:  
       a = 0  
       for t in ''.join(text.lower().split('\n')).split(' '):  
           if w == t:  
               a += 1  
           answer[w] = a  
   return answer

checkio - 08\_Bigger\_Price

//보경

def bigger\_price(limit: int, data: list) -> list:  
   sorted\_data = sorted(data, key=lambda x:x['price'], reverse=True)  
   return sorted\_data[:limit]

//민주

def bigger\_price(n, price):  
   answer = []  
   price\_only = [x['price'] for x in price]  
   for i in range(n):  
       answer.append(price[price\_only.index(sorted(price\_only, reverse=True)[i])])  
   return answer

//예슬

value\_list = []  
   max\_list = []  
   for i in range(len(data)):  
       value\_list.append(data[i]['price'])   value\_list.sort(reverse = True)   for i in range(limit):  
       if data[i]['price'] == value\_list[i]:  
           max\_list.append(data[i]['price'])   return max\_list

checkio - 09\_Fizz\_Buzz

//보경

def checkio(number: int) -> str:  
   if number % 15 == 0:  
       return 'Fizz Buzz'  
   if number % 3 == 0:  
       return 'Fizz'  
   if number % 5 == 0:  
       return 'Buzz'  
   else:  
       return str(number)     #answer는 string으로

//민주

def checkio(n):  
   answer = str(n)  
   if n % 15 == 0:  
       answer = 'Fizz Buzz'  
   if n % 5 != 0 and n % 3 == 0:  
       answer = 'Fizz'  
   if n % 3 != 0 and n % 5 == 0:  
       answer = 'Buzz'  
   return answer

//예슬

if number % 15 == 0:  
       return 'Fizz Buzz'  
   elif number % 3 == 0:  
       return 'Fizz'  
   elif number % 5 == 0 :  
       return 'Buzz'   return str(number)

Programmers - 완주하지못한 선수 코드 스레드

//민주

def solution(participant, completion):  
   answer = ''   participant.sort()  
   completion.sort()  
   for i in range(len(completion)):  
       if participant[i] != completion[i]:  
           answer = participant[i]  
           break  
       if answer == '':  
           answer = participant[-1]  
   return answer

//파키토샛

def solution(partis, completes):  
   partis.sort()  
   completes.sort()   for p, c in zip(partis, completes):  
       if p != c:  
           return p   return partis[-1]

Programmers - 모의고사 코드

//민주

def solution(answers):  
   answer = []  
   mark = [0, 0, 0]  
   # 1번 수포자 정답률  
   student1 = list(range(1, 6)) \* 2000  
   for i in range(len(answers)):  
       if answers[i] == student1[i]:  
           mark[0] += 1   # 2번 수포자 정답률  
   student2 = [2, 1, 2, 3, 2, 4, 2, 5] \* 1250   for i in range(len(answers)):  
       if answers[i] == student2[i]:  
           mark[1] += 1   # 3번 수포자 정답률  
   student3 = [3, 3, 1, 1, 2, 2, 4, 4, 5, 5] \* 1000   for i in range(len(answers)):  
       if answers[i] == student3[i]:  
           mark[2] += 1   max\_mark = max(mark)  
   for i in range(3):  
       if max\_mark == mark[i]:  
           answer.append(i+1)  
   return answer

the most numbers thread

//민주

def checkio(\*args):  
    if args:  
        answer = max(args) - min(args)  
        if type(max(args)) == float:  
            return round(answer, 3)  
        return answer  
  
    return 0

//보경

def checkio(\*args):  
    if len(args) > 0:  
        max\_num = max(args)  
        min\_num = min(args)  
        return max\_num - min\_num  
    else:  
        return 0

// 예슬

if len(args) > 0:  
        return max(args) - min(args)  
    else:  
        return 0

//키토

def checkio(\*args):  
    return max(args) - min(args) if args else 0

even the last thread

//보경

def checkio(array):  
    if len(array) > 0:  
        count = 0  
        last = array[len(array)-1]  
        for i in range(0, len(array)):  
            if i % 2 == 0:  
                count += array[i]  
        return last \* count  
    else:  
        return 0

//민주

def checkio(args):  
    answer = 0  
    if args:  
        for i in range(len(args)):  
            if i % 2 == 0:  
                answer += args[i]  
        answer \*= args[-1]  
    return answer

//예슬

num = 0  
    if array:  
        for i in range(len(array)):  
            if i%2==0:  
                num += array[i]  
        return num\*array[-1]  
  
    else:  
  
        return 0

//키토

def checkio(array):  
    if not array:  
        return 0  
    else:  
        return sum(n for i, n in enumerate(array) if i % 2 == 0) \* array[-1]

secret message thread

//민주

def find\_message(text):  
    answer = []  
    for word in text:  
        if word.isupper(): answer.append(word)  
    return ''.join(answer)

//보경

def find\_message(text: str) -> str:  
    message = ""  
    for i in text:  
        if i.isupper() == True:  
            message += i  
    return message

//키토

def find\_message(text: str) -> str:  
    return ''.join(c for c in text if c.isupper())

three words thread

//민주

def checkio(text):  
    text\_list = text.lower().split(' ')  
    answer = []  
    for i in range(len(text\_list)):  
        if ((text\_list[i].isupper() == False) and (text\_list[i].islower()) == False):  
            answer.append(i)  
    for i in answer:  
        if i < 3:  
            return False  
    return True

//보경

def checkio(words: str) -> bool:  
    split\_words = words.split()  
    count = 0  
    for i in split\_words:  
        # i가 알파벳 문자일때  
        if i.isalpha() == True:  
            count += 1  
        # 문자가 연속으로 나와야하니까 숫자가 나오면 count를 리셋  
        else:  
            count = 0  
        # count에 대한 if문을 for문 안에 넣을 생각을 못 해서 오래 걸렸음..  
        if count >= 3:  
            return True  
    return False

//예슬

def checkio(words: str) -> bool:  
   count = 0  
   words = words.split()  
   for i in words:  
       if i.isalpha():  
           count+=1  
       else:  
           count = 0  
       if count>=3:  
           return True  
  
   return False

// 키토

def checkio(words: str) -> bool:  
    words = words.split()  
    for i in range(len(words) - 2):  
        if words[i].isalpha() and words[i+1].isalpha() and words[i+2].isalpha():  
            return True  
    return False

index power thread

//민주

def power(array: list, n: int):  
    if len(array) < n:  
        return -1  
    return array[n] \*\* n

//보경

def index\_power(array: list, n: int) -> int:  
    if len(array)-1 < n :  
        return -1  
    else:  
        return pow(array[n], n)   # pow(n, m) : n의 m승

//예슬

if len(array) > n:  
       return array[n]\*\*n  
   else:  
       return -1

//키토

def index\_power(array: list, n: int) -> int:  
    return -1 if len(array) <= n else array[n] \*\* n

programmers k 번째 수

//민주

def solution(array, commands):  
    answer = [0] \* len(commands)  
    for i in range(len(commands)):  
        answer[i] = sorted(array[commands[i][0] - 1 : commands[i][1]])[commands[i][2]-1]  
  
    return answer

//보경

def solution(array, commands):  
    answer = []  
    for i in commands:  
        start = i[0]  
        finish = i[1]  
        k = i[2]-1  
        answer.append(sorted(array[start-1:finish])[k])  
    return answer

// 키토

def solution(array, commands):  
    ans = []   
    for cmd in commands:  
        start, end, k = cmd  
        part = sorted(array[start-1:end])  
        ans.append(part[k-1])  
          
    return ans

programmers 체육복

// 민주

## 통과  
def solution(n, lost, reserve):  
  
    # 순서대로  
    lost.sort()  
    reserve.sort()  
  
    # 잃어버렸지만 여벌 있는 경우  
    still\_got\_clothes = set(lost) & set(reserve)  
    lost = list(set(lost) - still\_got\_clothes)  
    reserve = list(set(reserve) - still\_got\_clothes)  
  
    # # 앞 뒤 번호에 Reserve 가 있는지, 빌리면 Remove  
    for number in reserve:  
  
        if (number - 1) in lost:  
            lost.remove(number - 1)  
  
        elif (number + 1) in lost:  
            lost.remove(number + 1)  
  
    return n - len(lost)

// 키토

def solution(N, lost, reserve):  
    lost, sorted\_reserve = sorted(lost), sorted(reserve)  
    reserve = []  
    # 문제 그 어디에도 목록이 정렬되어 있을 것이라는 희망을 품게 하는 문장은 없습니다.  
  
    for r in sorted\_reserve:  
        if r in lost:  
            lost.pop(lost.index(r))  
        else:  
            reserve.append(r)  
      
    for r in reserve:  
        if (r-1) in lost:  
            lost.pop(lost.index(r-1))  
        elif (r+1) in lost:  
            lost.pop(lost.index(r+1))  
      
    return N - len(lost)

프로그래머스 2016년

// 예슬

def solution(a, b):  
    answer = ''  
    month = [31, 29, 31, 30, 31, 30, 31, 31, 30, 31, 30, 31]  
    day = ['FRI', 'SAT', 'SUN', 'MON', 'TUE', 'WED', 'THU']  
    month\_sum = 0  
    for num in range(a-1):  
        month\_sum += month[num]  
  
    day\_num = (month\_sum + b) % 7  
    answer = day[day\_num - 1]  
  
    return answer

//지원

def solution(a, b):  
    enum = {0: 'SUN', 1: 'MON', 2: 'TUE', 3: 'WED', 4: 'THU', 5: 'FRI', 6: 'SAT'}  
    enumMonth = [31, 29, 31, 30, 31, 30, 31, 31, 30, 31, 30, 31]  
  
    x = 0  
    for i in range(0, a-1):  
        x += enumMonth[i]  
    return enum.get((x+b-3)%7)  
  
print(solution(5, 24))

//민주

def solution(a, b):  
    import pandas as pd  
    date\_form = '2016-{}-{}'  
    input\_date = pd.to\_datetime(date\_form.format(a, b))  
    day\_of\_week = ['MON', 'TUE', 'WED', 'THU', 'FRI', 'SAT', 'SUN']  
    return day\_of\_week[input\_date.weekday()]

//키토

DAYS = ['THU', 'FRI', 'SAT', 'SUN', 'MON', 'TUE', 'WED']  
MONTH\_DAYS = [0, 31, 29, 31, 30, 31, 30,  
                 31, 31, 30, 31, 30, 31]  
  
CUM\_MONTH\_DAYS = [0] \* 13  
for m in range(2, len(CUM\_MONTH\_DAYS)):  
    CUM\_MONTH\_DAYS[m] = CUM\_MONTH\_DAYS[m-1] + MONTH\_DAYS[m-1]  
# cum\_month\_days[m] : 1월부터 m-1월까지의 일의 누적합  
# [0, 0, 31, 60, 91, 121, 152, 182, 213, 244, 274, 305, 335]  
  
def solution(month, day):  
    days = CUM\_MONTH\_DAYS[month] + day  
    return DAYS[days % 7]

프로그래머스 가운데 글자 가져오기

//예슬

def solution(s):  
    answer = ''  
    lens = len(s)  
    if lens%2 == 0:  
        answer = s[(lens//2) - 1: (lens//2) + 1]  
        return answer  
    else :  
        answer = s[(lens//2)]  
        return answer

// 보경

def solution(s):  
    answer = ''  
    if len(s) % 2 != 0:  
        center = len(s) // 2  
        return answer.join(s[center])  
    else:  
        center\_first = (len(s) // 2)-1  
        center\_second = len(s) // 2  
        return answer.join([s[center\_first], s[center\_second]])

//지원

def solution(s):  
    answer = ''  
    length = len(s)  
    if length % 2 == 0:  
        answer+= s[length//2-1:length//2+1]  
    else:  
        answer+= s[length//2]  
  
    return answer

// 민주

def solution(s):  
    answer = ''  
  
    if len(s) % 2 == 1:  
        answer = s[len(s) // 2]  
  
    else:  
        answer = s[int(len(s) / 2 - 1):int(len(s) / 2 + 1)]  
    return answer

//키토

def solution(s):  
    MID = len(s) // 2  
    return s[MID] if len(s) % 2 == 1 else s[MID-1:MID+1]

checkio  #15 Right to Left

//보경

def left\_join(phrases):  
    phrase = ','.join(phrases).replace('right', 'left')  
    return phrase

//지원

def left\_join(phrases):  
    return {','.join(phrases)}.replace('right','left')

//민주

def left\_join(phrases):  
    return ''.join(phrases).replace('left', 'right')

//예슬

def left\_join(phrases):  
    phrase = ','.join(phrases).replace('right', 'left')  
    return phrase

//키토

def solution(words):  
    ans = ', '.join(words)  
    return ans.replace('right', 'left')

checkio  #16 Digits Multiplication

// 예슬

def checkio(number: int) -> int:  
    number = str(number)  
    multi = 1  
    for num in number:  
        if int(num) == 0:  
            pass  
        else:  
            multi = multi \* int(num)  
    return multi

//보경

def checkio(number: int) -> int:  
    mul = 1  
    for num in str(number):  
        if num != '0':  
            mul \*= int(num)  
    return mul

//민주

def checkio(number):  
    num\_list = [x for x in str(number) if x != '0']  
    answer = 1  
    for num in num\_list:  
        answer \*= int(num)  
  
    return answer

//키토

def solution(N):  
    ans = 1  
  
    while N:  
        N, rest = divmod(N, 10)  
        if rest == 0:  
            rest = 1  
        ans \*= rest

checkio  #17 Number Base

// 키토

from string import digits, ascii\_uppercase  
  
NUMBERS = digits + ascii\_uppercase  
  
def solution(N, base):  
    if NUMBERS.index(max(N)) >= base:  
        return -1  
  
    ans = 0  
    N = N[::-1]  
  
    for e, n in enumerate(N):  
        ans += NUMBERS.index(n) \* (base \*\* e)  
  
    return ans

checkio  #18 Absolute Sorting

//예슬

def checkio(numbers\_array: tuple) -> list:  
    a = []  
    abs\_numbers = list(numbers\_array)  
      
    for i in range(len(numbers\_array)):  
        a.append(0)  
        abs\_numbers[i] = abs(abs\_numbers[i])  
          
    abs\_numbers.sort()  
      
    for num in numbers\_array:  
        for i in range(len(a)):  
            if abs(num) == abs\_numbers[i]:  
                a[i] = num  
      
    return a

//보경

def checkio(numbers\_array: tuple) -> list:  
    sorted\_array = sorted(numbers\_array, key = abs)  
    return sorted\_array

//민주

def checkio(numbers\_array: tuple) -> list:  
    numbers = [x for x in numbers\_array]  
    numbers\_abs = sorted([abs(x) for x in numbers])  
#     [-x if x not in numbers else x for x in numbers\_abs]  
    answer = []  
    for num in numbers\_abs:  
        if num not in numbers:  
            answer.append(-num)  
        else: answer.append(num)  
    return answer

// 키토

def solution(arr):  
    return sorted(arr, key=lambda x: abs(x))

a = [0]\*10  
print(a) ---> [0,0,0,0,0,0,0,0,0,0]

checkio  #19 The Most Frequent

//보경

def most\_frequent(data: list) -> str:  
    fre = max(data, key = data.count)  
    return fre

//민주

def most\_frequent(array):  
    array\_ind = list(set(array))  
    answer = {}  
    for element in array\_ind:  
        answer[element] = array.count(element)  
    return max(answer, key=answer.get)

//키토

def solution(chars):  
    unique\_chars = set(chars)  
    tmp\_count = 0  
    ans = None  
  
    for char in unique\_chars:  
        if chars.count(char) > tmp\_count:  
            tmp\_count = chars.count(char)  
            ans = char  
  
    return ans

또다른 아이디어!!  
정렬 후 element 하나씩 보면서 max string과 갯수를 저장하도록!!

checkio Easy\_Unpack

//예슬

def easy\_unpack(elements: tuple) -> tuple:  
     
    numbers = (elements[0], elements[2], elements[-2])  
    return numbers

//민주

def easy\_unpack(elements):

answer\_index = [0, 2, -2]

return tuple([elements[x] for x in answer\_index])

checkio House\_Password

//민주

def checkio(data):

if len(data) >= 10:

if sum(map(str.isdigit, data)):

if sum(map(str.islower, data)):

if sum(map(str.isupper, data)):

return True

return False

checkio All the Same

//민주

def all\_the\_same(elements):

from collections import Counter

return len(Counter(elements)) < 2

checkio Monkey\_Typing

// 민주

# List Comprehension

def count\_words(text, words):

​

return sum([1 if x in text.lower() else 0 for x in words])

checkio Non-unique Elements

//민주

# collections Counter

def checkio(data):

​

from collections import Counter

answer = {}

for k, v in Counter(data).items():

if v > 1:

answer[k] = v

return [x for x in data if x in list(answer.keys())]

프로그래머스-  나누어 떨어지는 숫자 배열

//민주

def solution(arr, divisor):

answer = sorted([x for x in arr if x % divisor == 0])

if answer: return answer

return [-1]

//예슬

def solution(arr, divisor):      
    answer = sorted([i for i in arr if i % divisor == 0] )  
    if len(answer) == 0:  
        return [-1]  
    return answer

프로그래머스- 같은 숫자는 싫어

//민주

def no\_continuous(s):

a = []

for i in s:

if a[-1:] == [i]: continue

a.append(i)

return a

//예슬

def solution(arr):  
      
    answer = []  
    for i in range(len(arr)):  
        if i == 0:  
            answer.append(arr[i])  
        elif arr[i] != arr[i-1]:  
            answer.append(arr[i])  
  
    return answer

checkio Common Words

//민주

def checkio(text1, text2):

all\_text = (text1 + ',' + text2).split(',')

​

count\_word = {}

​

for word in all\_text:

count\_word[word] = all\_text.count(word)

​

max\_count = max(count\_word.values())

max\_index = []

for i, x in enumerate(list(count\_word.values())):

if x == 2:

max\_index.append(i)

return ','.join(sorted([list(count\_word.keys())[x] for x in max\_index]))

//보경

def checkio(first, second):

word1 = sorted(first.split(','))

word2 = sorted(second.split(','))

answer = ""

for i in range(0, len(word1)):

for j in range(0, len(word2)):

if word1[i] == word2[j]:

answer += word1[i] + ','

else:

pass

return answer[:-1]

//예슬

def checkio(first, second):  
    answer = ""  
    l = []  
    first = first.split(',')  
    second = second.split(',')  
    for i in first:  
        if i in second:  
            l.append(i)  
              
    l = sorted(l)  
      
    for i in range(len(l)):  
        if l[i] == l[-1]:  
            answer += l[i]  
        else :  
            answer += l[i] + ','  
  
    return answer

checkio Verify Aangrams

//민주

def verify\_anagrams(text1, text2):

text1 = sorted(text1.replace(' ', '').lower())

text2 = sorted(text2.replace(' ', '').lower())

​

return text1 == text2

//예슬

def verify\_anagrams(first\_word, second\_word):  
    first, second = sorted(''.join(first\_word.split()).lower()), sorted(''.join(second\_word.split()).lower())  
  
    return first == second

checkio Conversion from CamelCase

//민주

def from\_camelcase(text):

​

answer = ''

for i, x in enumerate(text):

if x.isupper(): answer += '\_'

answer += x.lower()

return answer[1:]

프로그래머스 두 정수 사이의 합

//민주

def solution(a, b):

​

answer = 0

for i in range(min(a, b), max(a, b) + 1):

answer += i

return answer

# Comprehension

def solution(a, b):

​

return sum([x for x in range(min(a, b), max(a, b) + 1)])

//보경

def solution(a, b):

if a > b:

a, b = b, a

if a == b:

return a

else:

answer = 0

for i in range(a, b+1):

answer += i

return answer

//예슬

def solution(a, b):  
    num\_list = sorted([a, b])  
    answer = 0  
      
    while num\_list[0] <= num\_list[1]:  
        answer += num\_list[0]  
        num\_list[0] += 1  
          
    return answer

Checkio Median

//민주

def checkio(array):

array.sort()

mid = len(array) // 2

if len(array) % 2 == 0:

return sum(array[mid - 1 : mid + 1])/2

return array[mid]

//예슬

def checkio(data: List[int]) -> [int, float]:  
    data = sorted(data)  
      
    if len(data)%2 == 0:  
        answer = (data[(len(data)//2) -1] + data[len(data)//2])/2  
    else :  
        answer = data[len(data)//2]  
      
    return answer

checkio conversion-into-camelcase

//예슬

def to\_camel\_case(name):  
    str\_list = name.split('\_')  
    for i in range(len(str\_list)):  
        str\_list[i] = str\_list[i][0].upper() + str\_list[i][1:]  
          
    answer = ''.join(str\_list)  
      
    return answer

//민주

def to\_camelcase(text):  
    text\_list = text.split('\_')  
      
    answer = ''  
    for word in text\_list:  
        answer += word[0].upper()  
        answer += word[1:]  
    return answer