💬 Problem

정수 N이 입력되면 00시 00분 00초부터 N시 59분 59초까지의 모든 시각 중에서 3이 하나라도 포함되는 모든 경우의 수를 구하는

프로그램을 작성하시오. 예를 들어 1을 입력했을 때 다음은 3이 하나라도 포함되어 있으므로 세어야 하는 시각이다.

- 00시 00분 03초

- 00시 13분 30초

반면에 다음은 3이 하나도 포함되어 있지 않으므로 세면 안 되는 시각이다.

- 00시 02분 55초

- 01시 27분 45초

코드

N = int(input())

c = 0

for i in range(N +1):

for j in range(60):

for k in range(60):

if '3' in str(i) + str(j) + str(k):

c += 1

print(c)

===============================================================================================================================================================

💬 Problem

한 마을에 모험가가 N명 있습니다. 모험가 길드에서는 N명의 모험가를 대상으로 '공포도'를 측정했는데, '공포도'가 높은 모험가는 쉽게 공포를 느껴 위험 상황에서 제대로 대처할 능력이 떨어집니다. 모험가 길드장인 동빈이는 모험가 그룹을 안전하게 구성하고자 공포도가 X인 모험가는 반드시 X명 이상으로 구성한 모험가 그룹에 참여해야 여행을 떠날 수 있도록 규정했습니다. 동빈이는 최대 몇개의 모험가 그룹을 만들 수 있는지 궁금합니다. 동빈이를 위해 N명의 모험가에 대한 정보가 주어졌을 때, 여행을 떠날 수 있는 그룹 수의 최댓값을 구하는 프로그램을 작성하세요.

⚠️ Limit

시간 제한 : 1초

첫째 줄에 모험가의 수 N이 주어집니다. (1 <= N <= 100,000)

둘째 줄에 각 모험가의 공포도의 값을 N 이하의 자연수로 주어지며, 각 자연수는 공백으로 구분합니다.

코드

import sys

n = int(sys.stdin.readline())

lst = list(map(int,sys.stdin.readline().split()))

groub = 0

c = max(lst) + 1

for i in range(1, c):

if i <= lst.count(i):

groub += (lst.count(i) // i)

print(groub)

===============================================================================================================================================================

💬 Problem

각 자리가 숫자(0부터 9)로만 이루어진 문자열 S가 주어졌을 대, 왼쪽부터 오른쪽으로 하나씩 모든 숫자를 확인하며 숫자 사이에 'x' 혹은 '+' 연산자를 넣어 결과적으로 만들어질 수 있는 가장 큰 수를 구하는 프로그램을 작성하세요. 단, +보다 x를 먼저 계산하는 일반적인 방식과는 달리 모든 연산은 왼쪽에서부터 순서대로 이루어진다고 가정합니다. 예를 들어 02984라는 문자열이 주어지면, 만들어질 수 있는 가장 큰수는 ((((0 + 2) \* 9) \* 8) \* 4) = 576입니다. 또한 만들어질 수 있는 가장 큰 수는 항상 20억 이하의 정수가 되도록 입력이 주어집니다.

⚠️ Limit

시간 제한 : 1초

첫째 줄에 여러 개의 숫자로 구성된 하나의 문자열 S가 주어집니다. (1 <= S의 길이 <= 20)

코드

n = input()

res = 0

for i in range(len(n)):

num = int(n[i])

if num == 0 or num == 1:

res += num

elif res == 0 and num > 1:

res += num

elif num > 1:

res \*= num

print(res)

===============================================================================================================================================================

💬 Problem

A, B 두 사람이 볼링을 치고 있습니다. 두 사람은 서로 무게가 다른 볼링공을 고르려고 합니다. 볼링공은 총 N개가 있으며 각 볼링공마다 무게가 적혀 있고, 공의 번호는 1번부터 순서대로 부여됩니다. 또한 같은 무게의 공이 여러 개 있을 수 있지만, 서로 다른 공으로 간주합니다. 볼링공의 무게는 1부터 M까지의 자연수 형태로 존재합니다. 예를 들어 N이 5이고, M이 3이며 각각의 무게가 차례대로 1, 3, 2, 3, 2일 때 각 곡의 번호가 차례대로 1번부터 5번까지 부여됩니다. 이때 두 사람이 고를 수 있는 볼링공 번호의 조합을 구하면 다음과 같습니다.

(1번, 2번), (1번, 3번), (1번, 4번), (1번, 5번), (2번, 3번), (2번, 5번), (3번, 4번), (4번, 5번)

결과적으로 두 사람이 공을 고르는 경우의 수는 8가지입니다. N개의 공의 무게가 각각 주어질 때, 두 사람이 볼링공을 고르는 경우의 수를 구하는 프로그램을 작성하세요.

⚠️ Limit

시간 제한 : 1초

첫째 줄에 볼링공의 개수 N, 공의 최대 무게 M이 공백으로 구분되어 각각 자연수 형태로 주어집니다. (1 <= N <= 1,000, 1 <= M <= 10)

둘째 줄에 각 볼링공의 무게 K가 공백으로 구분되어 순서대로 자연수 형태로 주어집니다. (1 <= K <= M)

코드

import sys

N, M= map(int,sys.stdin.readline().split())

K = list(map(int,sys.stdin.readline().split()))

count = 0

for i in range(N):

for j in K[i+1:]:

if K[i] != j:

count += 1

print(count)

===============================================================================================================================================================

💬 Problem - 풀지 못함

평소 식욕이 왕성한 무지는 자신의 재능을 뽐내고 싶어졌고 고민 끝에 카카오ㅓ TV 라이브 방송을 하기로 마음 먹었습니다. 그냥 먹방을 하면 다른 방송과 차별성이 없기 때문에 무지는 다음과 같이 독특한 방식을 생각해냈습니다. 회전판에 먹어야 할 N개의 음식이 있습니다. 각 음식에는 1부터 N까지 번호가 붙어 있으며, 각 음식을 섭취하는데 일정 시간이 소요됩니다. 무지는 다음과 같은 방법으로 음식을 섭취합니다. 1. 무지는 1번 음식부터 먹기 시작하며, 회전판은 번호가 증가하는 순서대로 음식을 무지 앞으로 가져다 놓습니다. 2. 마지막 번호의 음식을 섭취한 후에는 회전판에 의해 다시 1번 음식이 무지 앞으로 옵니다. 3. 무지는 음식 하나를 1초 동안 섭취한 후 남은 음식은 그대로 두고, 다음 음식을 섭취합니다. 다음 음식이란, 아직 남은 음식 중 다음으로 섭취해야 할 가장 가까운 번호의 음식을 말합니다. 4. 회전판이 다음 음식을 무지 앞으로 가져오는 데 걸리는 시간은 없다고 가정합니다. 무지가 먹방을 시작한 지 K초 후에 네트워크 장애로 인해 방송이 잠시 중단 되었습니다. 무지는 네트워크 정상화 후 다시 방송을 이어갈 때, 몇 번 음식부터 섭취해야 하는지를 알고자 합니다. 각 음식을 모두 먹는데 필요한 시간이 담겨 있는 배열 food\_times, 네트워크 장애가 발생한 시간 K초가 매개변수로 주어질 때 몇 번 음식부터 다시 섭취하면 되는지 return 하도록 solution 함수를 완성하세요.

⚠️ Limit

시간 제한 : 1초

food\_times는 각 음식을 모두 먹는 데 필요한 시간이 음식의 번호 순서대로 들어 있는 배열입니다.

k는 방송이 중단된 시간을 나타냅니다.

만약 더 섭취해야 할 음식이 없다면 -1을 반환하면 됩니다.

food\_times의 길이는 1 이상 2,000 이하입니다.

food\_times의 원소는 1 이상 1,000 이하의 자연수입니다.

k는 1이상 2,000,000 이하의 자연수입니다.

food\_times의 길이는 1 이상 200,000 이하입니다.

food\_times의 원소는 1 이상 100,000,000 이하의 자연수입니다.

k는 1 이상 2 x 1013 이하의 자연수입니다.

코드

def solution(food\_time, K):

N = K // len(food\_time)

M = K % len(food\_time)

c = 0

for i in range(len(food\_time)):

food\_time[i] -= N

if food\_time[i] <= 0:

M += -(food\_time[i])

del food\_time[i]

res = 0

while M != 0:

for i in range(len(food\_time)):

food\_time[i] -= 1

M -= 1

if food\_time[i] == 0:

del food\_time[i]

===============================================================================================================================================================

💬 Problem

게임의 아웃복서 캐릭터는 필살기인 '럭키 스트레이트' 기술이 있습니다. 이 기술은 매우 강력한 대신에 게임 내에서 점수가 특정 조건을 만족할 때만 사용할 수 있습니다. 특정 조건이란 현재 캐릭터의 점수를 N이라고 할 때 자릿수를 기준으로 점수 N을 반으로 나누어 왼쪽 부분의 각 자릿수의 합과 오른쪽 부분의 각 자릿수의 합을 더한 값이 동일한 상황을 의미합니다. 예를 들어 현재 점수가 123,402라면 왼쪽 부분의 각 자릿수의 합은 1 + 2 + 3, 오른쪽 부분의 각 자릿수의 합은 4 + 0 + 2 이므로 두 합이 6으로 동일하여 럭키 스트레이트를 사용할 수 있습니다. 현재 점수 N이 주어지면 럭키 스트레이트를 사용할 수 있는 상티인지 아닌지를 알려주는 프로그램을 작성하세요.

⚠️ Limit

시간 제한 : 1초

첫째 줄에 점수 N이 정수로 주어집니다. (10 <= N <= 99,999,999) 단, 점수 N의 자릿수는 항상 짝수 형태로만 주어집니다. 예를 들어 자릿수가 5인 12,345와 같은 수는 입력으로 들어오지 않습니다.

코드

N = input()

half = len(N) // 2

right = 0

left = 0

for i in range(half):

left += int(N[i])

for i in range(half):

right += int(N[i+half])

if left == right:

print('LUCKY')

else:

print('READY')

===============================================================================================================================================================

💬 Problem

알파벳 대문자와 숫자 (0 ~ 9)로만 구성된 문자열이 입력으로 주어집니다. 이때 모든 알파벳을 오름차순으로 정렬하여 이어서 출력한 뒤에, 그 뒤에 모든 숫자를 더한 을 이어서 출력합니다. 예를 들어 K1KA5CB7이라는 값이 들어오면 ABCKK13을 출력합니다.

⚠️ Limit

시간 제한 : 1초

첫째 줄에 하나의 문자열 S가 주어집니다. (1 <= S의 길이 <= 10,000)

코드

from collections import deque

import sys

S = sys.stdin.readline()

alp = deque()

num = deque()

for i in range(len(S)-1):

if S[i] >= 'A' and S[i] <= 'Z':

alp.append(S[i])

else:

num.append(S[i])

for i in sorted(alp):

print(i,end='')

for i in sorted(num):

print(i,end='')