Министерство цифрового развития, связи

и массовых коммуникаций Российской Федерации

Ордена Трудового Красного Знамени

Федеральное государственное бюджетное образовательное учреждение высшего образования

«Московский технический университет связи и информатики»

Курсовая работа

по дисциплине «Структуры и алгоритмы обработки данных»

Выполнила:

студентка группы БСТ2004

Моисеева С.Д.

Проверил:

А.Д. Чайка

Москва 2022

**Задание 991C**

**Эклеры (1500 баллов)**

**Условие:**

После успешной сдачи всех зачетов Вася купил себе в подарок коробку, содержащую nn сладких эклеров. Вася решил каждое утро есть некоторое одинаковое число эклеров, пока они все не закончатся. Однако сосед Васи, Петя, заметил принесенную Васей коробку и тоже решил насладиться вкусом эклеров.

Теперь процесс поедания эклеров выглядит следующим образом: сначала Вася выбирает число kk, одинаковое для всех дней. Затем утром он съедает kk эклеров из коробки (или доедает все эклеры, если их осталось меньше kk), после этого Петя вечером съедает 10%10% **оставшихся** эклеров. Если эклеры еще не закончились, то на следующий день Вася опять съедает kk эклеров, а Петя — 10%10% от оставшихся и так далее.

Если число эклеров не делится на 1010, то Петя округляет «свою» долю в меньшую сторону, например, если в коробке было 9797 эклеров, то Петя съест только 99 из них. В частности, если в коробке уже меньше 1010 эклеров, то Петя не будет их есть вообще.

Определите, какое наименьшее число kk может выбрать Вася такое, что он съест не менее половины от всех nn эклеров, которые были в коробке изначально. Заметьте, что число kk должно быть натуральным.

**Входные данные**

В первой строке содержится натуральное число nn (1≤n≤10181≤n≤1018) — изначальное количество эклеров.

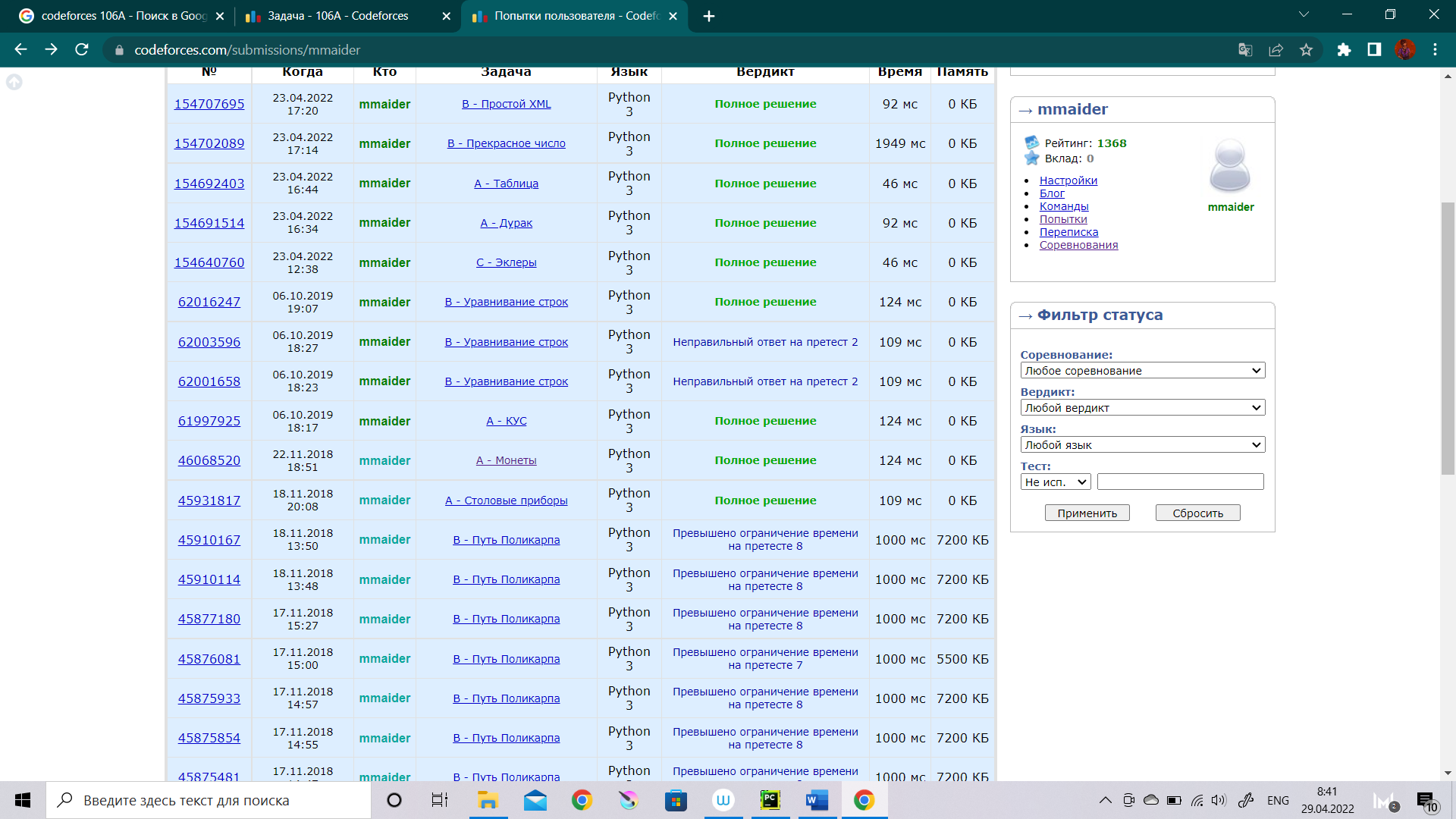
**Выходные данные**

Вывести единственное число — наименьшее значение kk, удовлетворяющее Васю.

**Код решения:**

def f(k):  
 s = m = n  
 while s > 0 and m:  
 l = min(k, m)  
 s -= 2 \* l  
 m -= l  
 m -= m // 10  
 return s <= 0  
  
  
n = int(input())  
l = [0, n]  
while l[1] - l[0] > 1:  
 m = sum(l) // 2  
 l[f(m)] = m  
**print(l[1])**

**Вердикт:**



**Задание 106A**

**Дурак (1000)**

**Условие:**

Дурак — карточная игра, популярная в странах бывшего СССР. В этой задаче не раскрывается всех правил этой игры — при желании вы можете найти их потом сами.

В дурака играют колодой из 36 карт. Каждая карта имеет масть («S», «H», «D» и «C») и достоинство (в порядке увеличения: «6», «7», «8», «9», «T», «J», «Q», «K» и «A»). В начале каждой партии одна из мастей случайным образом назначается *козырной*.

Ходы игроков состоят в том, что один игрок выкладывает на стол одну или несколько своих карт, а другой должен покрыть каждую из них своими картами.

Одна карта кроет другую если у них одинаковые масти, а достоинство первой выше достоинства второй. Кроме того, карта козырной масти может покрыть любую карту не козырной масти вне зависимости от их достоинств. Во всех остальных случаях вторую карту первой покрыть нельзя.

Вам даны козырная масть и две различные карты. Определите покрывает ли первая их них вторую или нет.

**Входные данные**

В первой строке находится козырная масть. Это «S», «H», «D» или «C».

Во второй строке находится описание двух различных карт. Каждая из карт описывается словом из двух символов. Первый символ означает достоинство («6», «7», «8», «9», «T», «J», «Q», «K» или «A»), а второй — масть («S», «H», «D» или «C»).

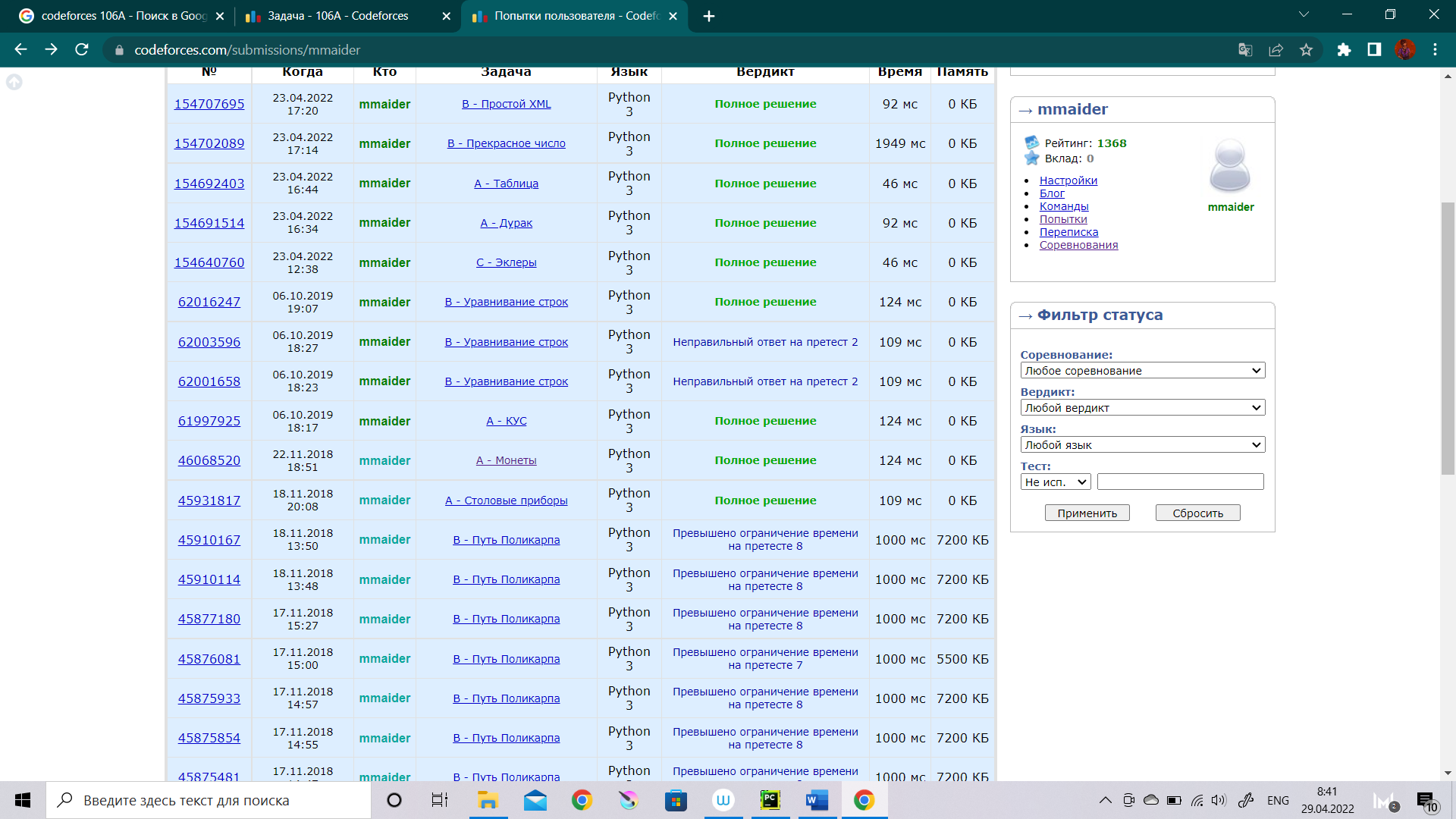
**Выходные данные**

Выведите «YES» (без кавычек) если первая карта покрывает вторую. Иначе выведите «NO» (тоже без кавычек).

**Код решения:**

t = '6789TJQKA'  
s = input()  
x = input()  
if x[1] == s and x[4] != s or x[1] == x[4] and t.find(x[0]) > t.find(x[3]):  
 print('YES')  
else:  
 print('NO')

**Вердикт:**



**Задание 359A**

**Таблица (1000)**

**Условие:**

У Семена есть прямоугольная таблица, состоящая из *n* строк и *m* столбцов. Семен пронумеровал строки таблицы, начиная от единицы, сверху вниз, а столбцы — начиная от единицы, слева направо. Ячейку, расположенную на пересечении *x*-ой строки и *y*-ого столбца, будем обозначать парой чисел (*x*, *y*). Углами таблицы будем называть ячейки: (1, 1), (*n*, 1), (1, *m*), (*n*, *m*).

Семен считает, что в этой таблице некоторые ячейки являются *хорошими*. Причем известно, что ни одна хорошая ячейка не является углом таблицы.

Изначально все ячейки таблицы бесцветны. Семен хочет покрасить все ячейки своей таблицы. За один ход он может выбрать любую хорошую ячейку таблицы (*x*1, *y*1), произвольный угол таблицы (*x*2, *y*2) и закрасить все ячейки таблицы (*p*, *q*), для которых выполняются оба неравенства: *min*(*x*1, *x*2) ≤ *p* ≤ *max*(*x*1, *x*2), *min*(*y*1, *y*2) ≤ *q* ≤ *max*(*y*1, *y*2).

Помогите Семену! Определите минимальное количество операций, необходимых ему для покраски всех ячеек таблицы. Обратите внимание, что одну ячейку разрешается покрасить несколько раз.

**Входные данные**

В первой строке заданы ровно два целых числа *n*, *m* (3 ≤ *n*, *m* ≤ 50).

Далее в *n* строках задано описание ячеек таблицы. А именно, в *i*-ой строке задано *m* целых чисел через пробел *ai*1, *ai*2, ..., *aim*. Если *aij* равно нулю, то ячейка (*i*, *j*) не является хорошей, иначе — *aij* равно единице. Гарантируется, что хотя бы одна ячейка является хорошей. Гарантируется, что ни одна хорошая ячейка не является углом.

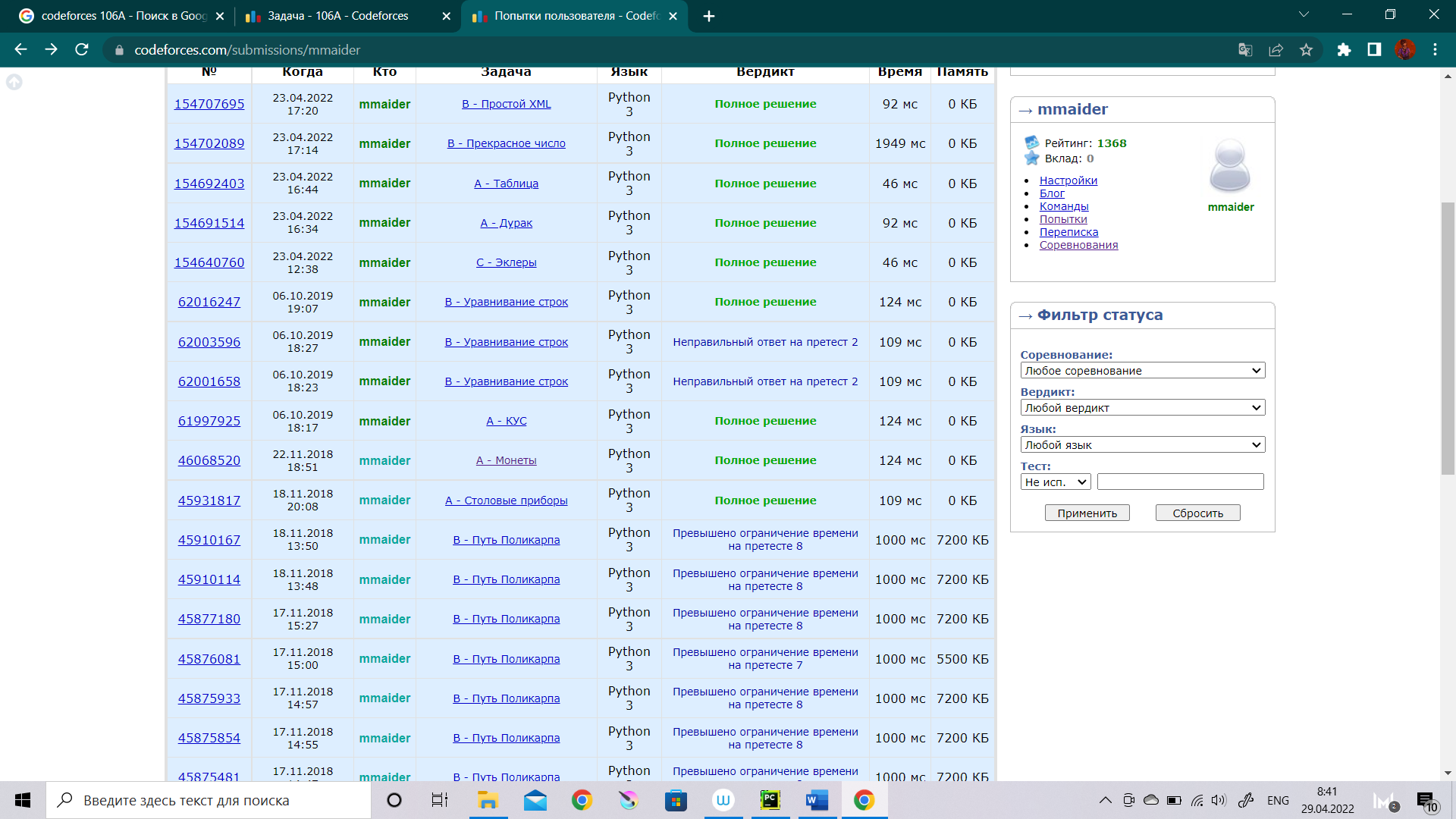
**Выходные данные**

Выведите единственное число — минимальное количество операций, необходимых Семену для осуществления его затей.

**Код решения:**

n, m = map(int, input().split())  
b = [list(map(int, input().split())) for i in range(n)]  
if 1 in b[0] or 1 in b[n - 1]:  
 print(2)  
else:  
 for i in range(n):  
 if b[i][0] == 1 or b[i][m - 1] == 1:  
 print(2)  
 break  
 elif i == n - 1:  
 print(4)

**Вердикт:**



**Задание 919B**

**Прекрасное число (1000)**

**Условие:**

Будем называть положительное целое число прекрасным, если и только если сумма его цифр равна 1010. Вам дано целое число kk, найдите kk-е по величине прекрасное положительное целое число.

**Входные данные**

Единственная строка содержит одно целое число kk (1≤k≤100001≤k≤10000).

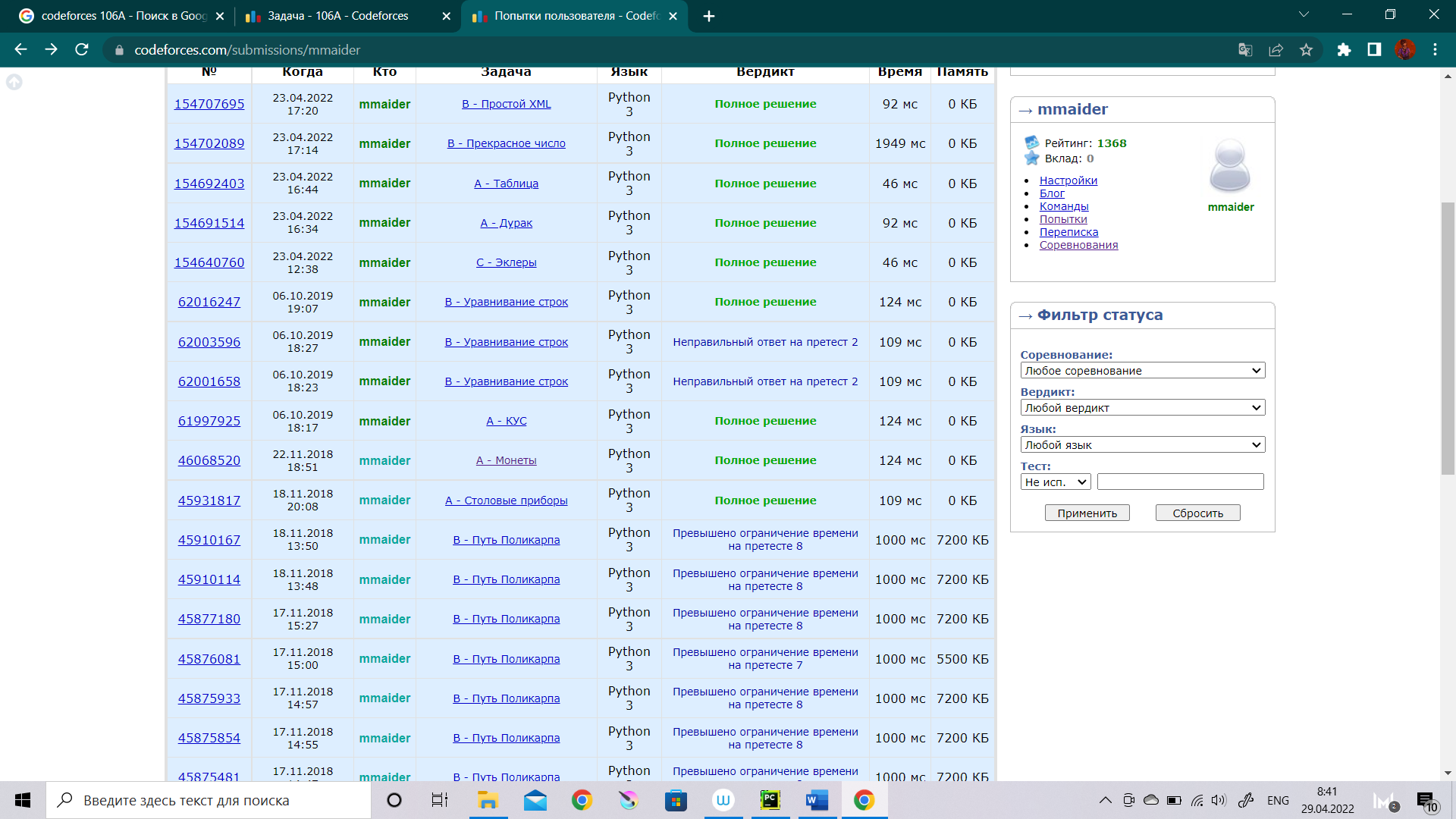
**Выходные данные**

Выведите kk-е по величине прекрасное число.

**Код решения:**

n = int(input())  
c = 10  
while n:  
 c += 9  
 if sum(map(int, str(c))) == 10:  
 n -= 1  
print(c)

**Вердикт:**



**Задание 125B**

**Прекрасное число (1000)**

**Условие:**

Назовем открывающим тегом строку <x>, где *x* — любая строчная буква латинского алфивита. Каждому открывающему тегу соответствует закрывающий тег вида </x>, где *x* — та же самая буква.

Теги могут вкладываться друг в друга: в этому случае одна пара из открывающего и закрывающего тега находится внутри другой пары.

Определим понятие *XML-текста*:

* пустая строка — это *XML-текст*
* если *s* — это *XML-текст*, то *s*'=<a>+*s*+</a> — тоже *XML-текст*, где *a* — любая строчная буква латинского алфавита
* если *s*1, *s*2 — это *XML-тексты*, то *s*1+*s*2 — тоже *XML-текст*

Вам дан XML-текст (гарантируется, что он корректен), ваша задача — вывести его в формате:

* каждый тег (как открывающий, так и закрывающий) находится на отдельной строке
* перед тегом выводится 2 \* *h* пробелов, где *h* — уровень вложенности тега.

**Входные данные**

Входные данные состоят из единственной непустой строки — XML-текста, его длина не превосходит 1000 символов. Гарантируется корректность этого текста. Текст не содержит пробелов.

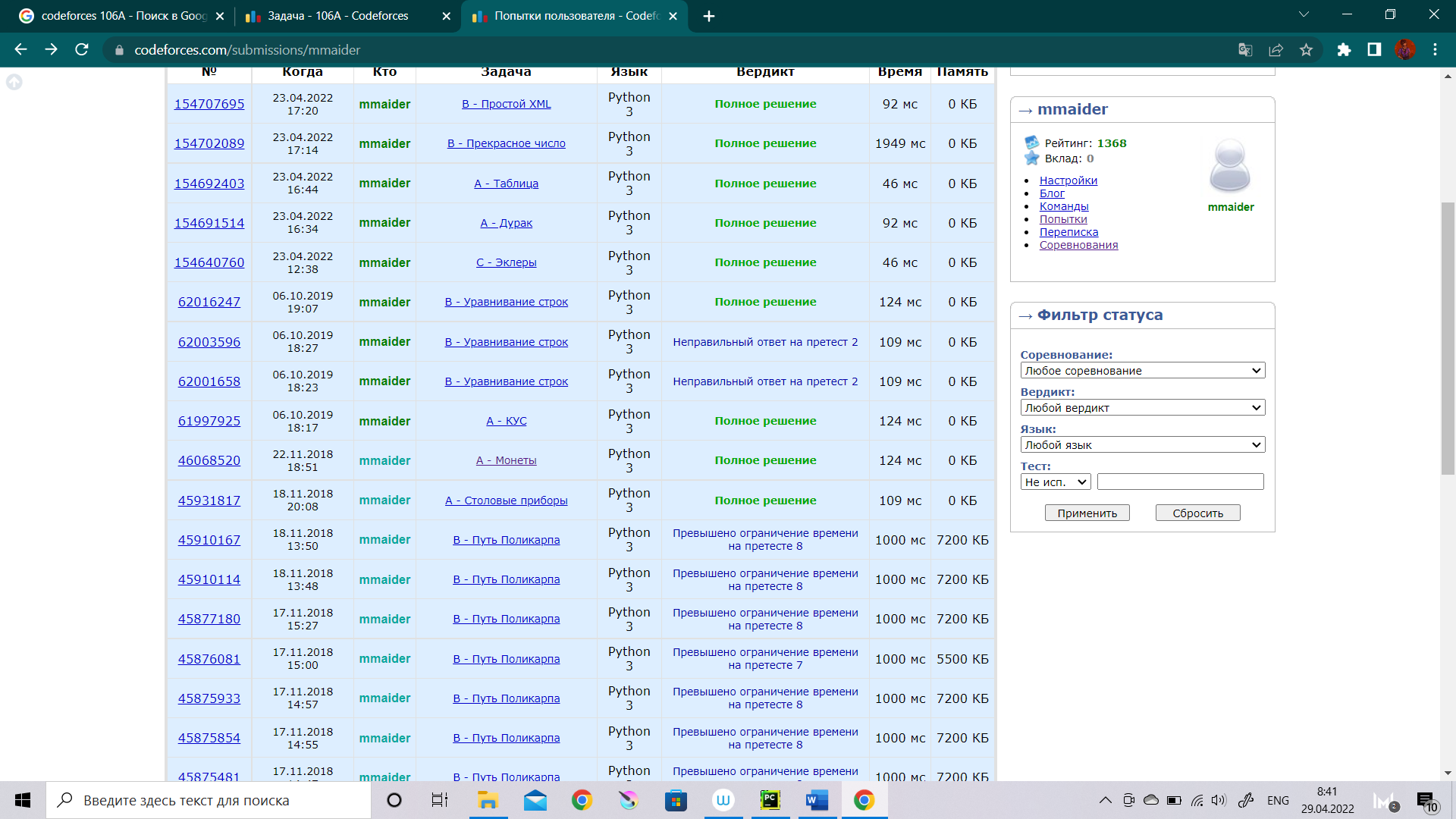
**Выходные данные**

Выведите данный XML-текст в соответствии с описанными выше правилами.

**Код решения:**

s = input()  
s = s[1: len(s) - 1].split("><")  
t = -1  
for i in s:  
 if i[0] != "/":  
 t += 1  
 print(" " \* t + "<" + i + ">")  
 else:  
 print(" " \* t + "<" + i + ">")  
 t -= 1

**Вердикт:**



**Решение на github: https://github.com/mmaider/codeforces**