Федеральное агентство связи

Ордена Трудового Красного Знамени

федеральное государственное бюджетное учреждение высшего образования

«Московский технический университет связи и информатики»

Кафедра «Математическая кибернетика и информационные технологии»

Дисциплина «Структуры и алгоритмы обработки данных»

Курсовая работа

Выполнила студентка группы БСТ2002

Сергеева А.А.

Проверил: Аршинов Е.А.

Москва 2022

**Задача 1560F2** (Сложность 2100)

Дано число n. Найдите минимальное целое число x такое, что x≥nx≥n и x является k-красивым числом.

Число называется k-красивым, если в его десятичной записи, не содержащей лидирующих нулей, встречается не более k различных цифр. Например, если k=2, числа 34344433434443, 5555055550, 777777 и 2121 являются kk-красивыми, а числа 120120, 445435445435 и 998244353998244353 — нет.

Входные данные

В первой строке записано одно целое число t (1≤t≤1041≤t≤104) — количество наборов входных данных. Далее следуют t наборов входных данных.

Каждый набор входных данных состоит из одной строки, содержащей два целых числа n и k (1≤n≤1091≤n≤109, 1≤k≤101≤k≤10).

Выходные данные

Для каждого набора входных данных в отдельной строке выведите x — минимальное целое k-красивое число, так чтобы x≥nx≥n.

**Решение**

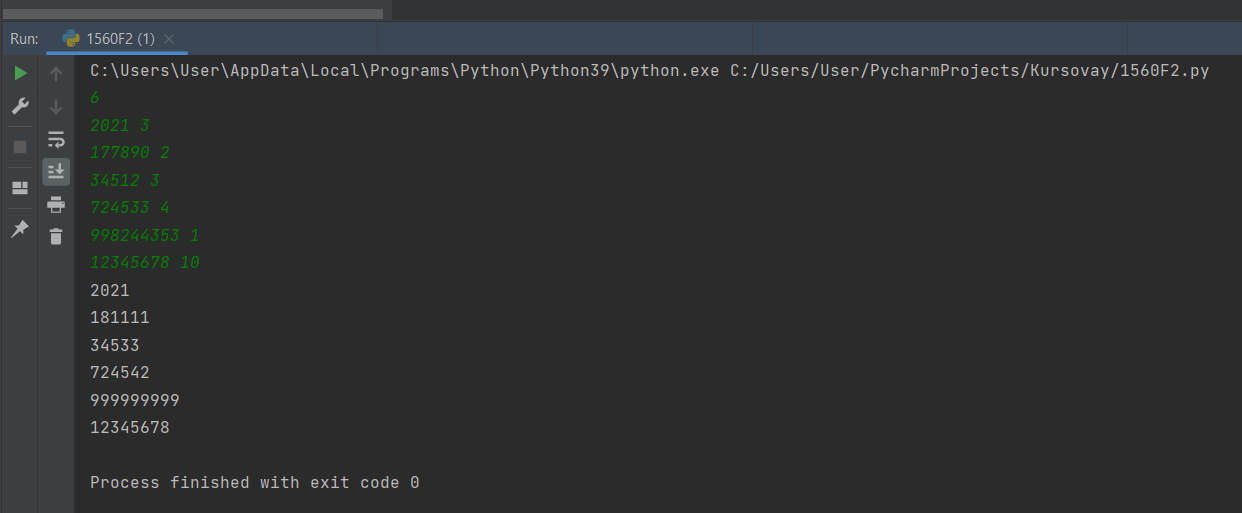
Суть заключается в том, что нужно найти наибольший префикс числа n, который содержит не более k различных цифр. Пусть этот префикс имеет длину p. Если p=m, то n уже является k-красивым, то выводим его. Если нет, то нужно увеличить префикс длины p+1 как число. Все остальные цифры заполним минимальными значениями, входящими в k различных цифр. Чтобы найти ответ для нового n, запустим описанную процедуру заново с новым значением n, пока число не будет иметь k различных цифр. В конце выводим минимальное найденное число.

**Код программы:**

t = int(input())  
some\_list = []  
for i in range(t):  
 some\_list.append(input())  
for i in range(t):  
 n\_k = some\_list[i].spl

it()  
 n = int(n\_k[0])  
 k = int(n\_k[1])  
 x = n  
 while len(set(str(n))) > k:  
 if n % 10 == 0:  
 n = n // 10  
 else:  
 n += 1  
 end = str(min(str(n))) \* (len(str(x)) - len(str(n)))  
 print(int(str(n) + end))

**Результат выполнения программы:**



**Задача 1526B** (сложность 1400)

Вам дано целое число x. Можете ли вы получить x, просуммировав некоторое количество 11, 111, 1111, 11111, …? (Вы можете использовать любое число среди них любое количество раз).

Например,

* 33=11+11+11
* 144=111+11+11+11

Входные данные

Первая строка ввода содержит одно целое число t (1≤t≤10000)— количество наборов входных данных.

Первая и единственная строка каждого набора входных данных содержит одно целое число x (1≤x≤))— число, которое вы должны получить.

Выходные данные

Для каждого набора входных данных вы должны вывести одну строку. Если вы можете получить x, выведите «YES» (без кавычек). В противном случае выведите «NO».

Вы можете вывести каждую букву из «YES» и «NO» в любом регистре (верхнем или нижнем).

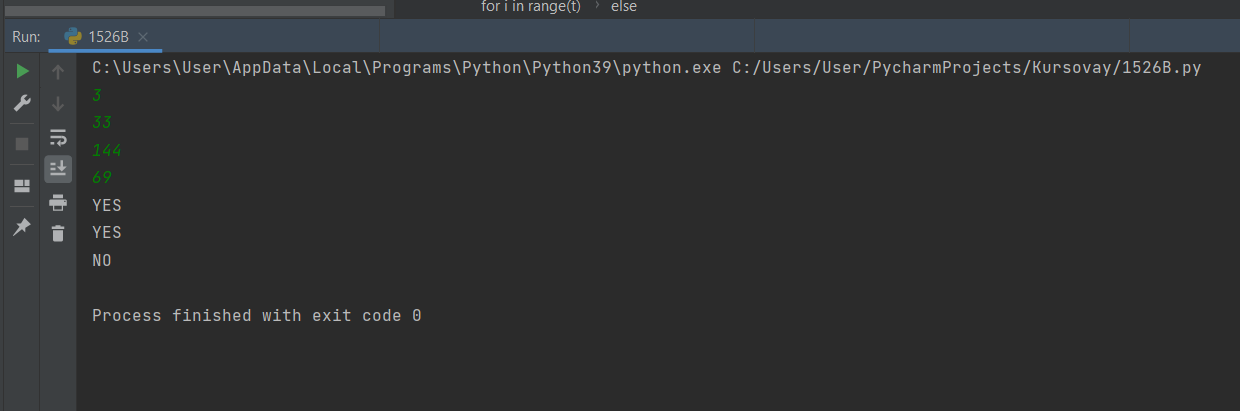
**Решение**

Суть заключается в том, что нужно сделать проверку на кратность 11 (остаток от деления должен быть равен 0), и в то же время, чтобы уменьшить время выполнения программы, добавляю в проверку еще 1 условие, которое проверяет, состоит ли число из 111, 1111 и т.д. Проверку кратности я умножаю на 111. Если наше число будет больше или равняться выражению (n % 11) \* 111, то наше число состоит из просуммированных числе 11, 111, 1111 и т.д. В конце выводим «YES», в противном случае выведите «NO».

**Код программы:**

t = int(input())  
for i in range(t):  
 n = int(input())  
 if n >= (n % 11) \* 111:  
 print('YES')  
 else:  
 print('NO')

**Результат выполнения программы:**



**Задача 1630C** (сложность 2200)

Дано n элементов, пронумерованных от 1 до n. Элемент i имеет значение ai и цвет ci. Изначально ci=0для всех i.

Можно выполнять следующую операцию:

Выбрать три элемента i, j и k (1≤i<j<k≤n) такие, что ci, cj и ck равны 0 и ai=ak, и затем присвоить cj=1.

Найдите максимальное значение ∑i=1nci, которое можно получить, выполнив описанную операцию некоторое (любое) количество раз.

Входные данные

Первая строка содержит целое число n (3≤n≤2⋅) — количество элементов.

Вторая строка содержит nn целых чисел a1, a2…, an (1≤ai≤n1≤ai≤n) — где ai равно значению i-го элемента.

Выходные данные

Напечатайте одно целое число — максимальное значение ∑i=1nci, которое можно получить, выполнив описанную операцию некоторое (любое) количество раз.

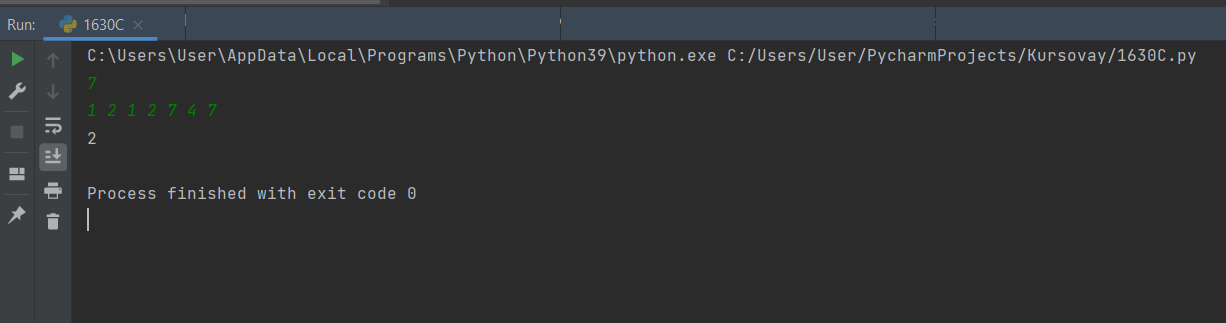
**Решение**

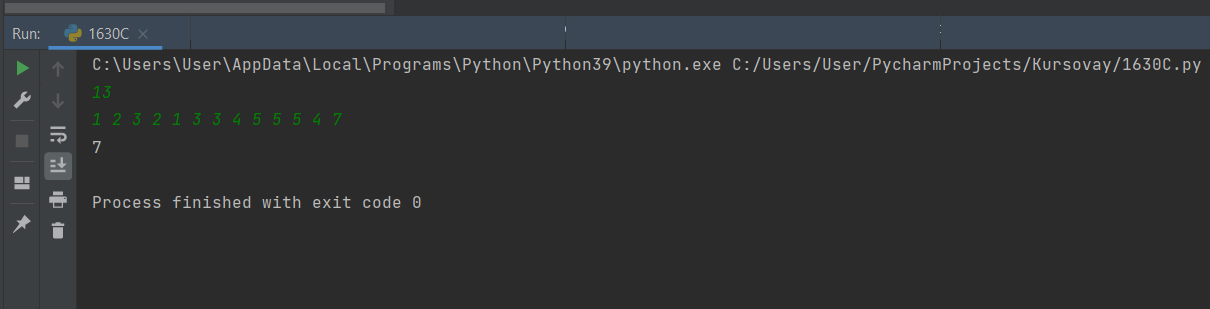
Суть заключается в том, что сначала я создаю словарь с индексами и значениями элемента (в паре ключ значение). Если у нас элементы повторяются, то в словаре происходит перезапись. Далее завожу 3 переменные, которым присваиваю 0: последний элемент; элемент, который «кроем» и счетчик. В цикле иду по элементам с 0-го индекса, используя сформированный словарь. В начале мы выбираем минимальное последнее число, между значением из словаря и тем, что мы присвоили переменной «последний элемент». Условие проверяет на то, чтобы индекс следующего элемента был обязательно больше текущего, если условие выполняется, то мы увеличиваем наш счетчик. Иначе мы элементу, который «кроем», присваиваем значение последнего. В конце выводим результат.

**Код программы:**

n = int(input())  
a = [int(i) for i in input().split()]  
  
last = {x: i for i, x in enumerate(a)}  
# print(last)  
  
lastest, covered, summa = 0, 0, 0  
for i, x in enumerate(a):  
 lastest = max(lastest, last[x])  
 if covered > i:  
 summa += 1  
 else:  
 covered = lastest  
  
print(summa)

**Результаты выполнения программы:**





**Заключение**

Все задачи прошли тестирования на сайте <https://codeforces.com>. Результаты представлены ниже.

