**Федеральное агентство связи**

**Ордена Трудового Красного Знамени**

**Федеральное государственное бюджетное образовательное учреждение высшего образования**

**«Московский технический университет связи и информатики»**

Кафедра “Математическая кибернетика и информационные технологии”

**Отчет по курсовой работе**

по дисциплине «Структуры и алгоритмы обработки данных»

Вариант №17.

Выполнила: студент группы БВТ2004

Ким Игорь Владимирович

Проверил:

Мкртчян Грач Маратович

Москва, 2022

# Задачи

## Задача №1

Перестановка — это последовательность длины n целых чисел от 1 до n, в которой все числа встречаются ровно по одному разу. Например, (1), (4, 3, 5, 1, 2), (3, 2, 1) — перестановки, а (1, 1), (4, 3, 1), (2, 3, 4) — нет.

Бывает много различных задач на перестановки. Сегодня вам предстоит решить такую. Представьте, что Некто взял несколько перестановок (возможно, с разным количеством элементов), записал их в одну последовательность одну за другой, а затем перемешал полученную последовательность. Требуется восстановить исходные перестановки, если это возможно.

Входные данные В первой строке содержится целое число n (1 ≤ n ≤ 105). Следующая строка содержит перемешанную последовательность из n целых чисел, разделенных одиночными пробелами. Числа в последовательности от 1 до 105.

Выходные данные Если данную последовательность можно разбить на несколько перестановок так, что каждый элемент последовательности принадлежит в точности одной перестановке, в первой строке выведите количество получившихся перестановок. Вторая строка должна содержать n чисел, соответствующих элементам заданной последовательности. Если i-й элемент относится к первой перестановке, то i-е число должно быть 1, если ко второй — то 2, и т.д. Порядок нумерации перестановок произволен.

Если решений несколько, выведите любое. Если решения не существует, выведите в первой строке  - 1. Примеры  
входные данные  
9  
1 2 3 1 2 1 4 2 5  
выходные данные  
3  
3 1 2 1 2 2 2 3 2

## Задача №2

В 2013-м году авторский коллектив Берляндского государственного университета должен подготовить задачи к n олимпиадам. Будем считать, что олимпиады пронумерованы последовательными целыми числами от 1 до n. Про каждую олимпиаду известно, сколько членов жюри должно участвовать в ее подготовке, а также время, требуемое на подготовку задач для нее. А именно, олимпиаду с номером i должны готовить pi человек в течение ti дней, причем подготовка олимпиады должна вестись непрерывный промежуток времени и закончиться ровно за день до олимпиады. Непосредственно в день олимпиады члены жюри, которые ее готовили, уже над ней не работают.

К примеру, если олимпиада состоится 9 декабря, и на ее подготовку требуется 7 человек и 6 дней, то все эти 7 членов жюри будут работать над задачами этой олимпиады с 3-го декабря по 8-е декабря (9-го декабря эти члены жюри уже не будут работать над задачами этой олимпиады, то есть некоторые из них могут начать готовить задачи какой-то другой олимпиады). А если олимпиада проводится 3 ноября и требует 5 дней подготовки, то члены жюри будут работать с 29-го октября по 2-е ноября.

Чтобы не перегружать членов жюри, было введено такое правило: один член жюри не может в один и тот же день работать над задачами для разных олимпиад. Напишите программу, которая определяет, какое наименьшее количество человек должно входить в состав жюри, чтобы все олимпиады могли быть подготовлены в срок.

Входные данные В первой строке записано целое число n — количество олимпиад в 2013-м году (1 ≤ n ≤ 100). Каждая из следующих n строк содержит по четыре целых числа mi, di, pi и ti — месяц и день олимпиады (заданы без лидирующих нулей), требуемое количество членов жюри и время, требуемое на подготовку i-й олимпиады (1 ≤ mi ≤ 12, di ≥ 1, 1 ≤ pi, ti ≤ 100), di не превосходит количества дней в месяце mi. Олимпиады заданы в произвольном порядке. В один день может проводиться несколько олимпиад.

Для решения задачи используйте современный (Григорианский) календарь. Обратите внимание, все даты заданы в 2013 году. Этот год не является високосным, следовательно, в феврале 28 дней. Обратите внимание, подготовка некоторой олимпиады может начаться в 2012 году.

Выходные данные Выведите единственное число — минимальный размер жюри.  
Примеры  
входные данные  
2  
5 23 1 2  
3 13 2 3  
выходные данные  
2

## Задача №3

Вася изучает позиционные системы счисления. К сожалению, он часто забывает записать основание системы счисления, в которой записано выражение. Однажды он увидел в своей тетради запись a + b = ?, при этом основание системы счисления нигде записано не было. Теперь Вася должен выбрать некоторое основание p и считать что выражение записано в p-ичной системе счисления. Вася понял, что при разных основаниях он может получать разные ответы, а некоторые основания даже недопустимы.

Например, выражение 78 + 87 в 16-ичной системе счисления равняется FF16, в 15-ичной: 11015, в 10-ичной: 16510, в 9-ичной: 1769, а в системах счисления с основанием 8 или меньше данное выражение недопустимо, т. к. все цифры должны быть строго меньше основания системы счисления. Васе стало интересно, какова длина самого длинного возможного значения выражения. Помогите ему найти эту длину.

Под длиной числа следует понимать количество цифр в нем. Например, длина самого длинного ответа для 78 + 87 = ? — это 3. Это получается, например, в 15-ичной (11015), 10-ичной (16510), 9-ичной (1769) и некоторых других системах счисления.

Входные данные В первой строке через пробел записано два числа a и b (1 ≤ a, b ≤ 1000) — заданные слагаемые.

Выходные данные Выведите одно число — длину наидлиннейшего ответа. Примеры  
входные данные  
78 87  
выходные данные  
3

## Задача №4

Красная шапочка нарисовала n прямоугольников. Углы прямоугольников имеют целочисленные координаты, а их ребра параллельны осям Ox и Oy. Прямоугольники могут касаться друг друга, но не могут пересекаться (то есть не существует точки, которая принадлежит внутренности более одного прямоугольника).

Ваша задача — определить, образуют ли нарисованные прямоугольники квадрат. Иными словами, определите, существует ли такой квадрат, что множество точек, лежащих внутри или на границе хотя бы одного прямоугольника, в точности равно множеству точек, лежащих внутри или на границе этого квадрата.

Входные данные Первая строка содержит целое число n (1 ≤ n ≤ 5). Каждая из следующих n строк содержит четыре целых числа, которые описывают один прямоугольник: x1, y1, x2, y2 (0 ≤ x1 < x2 ≤ 31400, 0 ≤ y1 < y2 ≤ 31400) — x1 и x2 обозначают x-координаты левой и правой стороны прямоугольника, y1 и y2 обозначают y-координаты нижней и верхней стороны прямоугольника.

Прямоугольники не пересекаются (то есть не существует точки, которая принадлежит внутренности бол ее одного прямоугольника).

Выходные данные В единственную строку выведите «YES», если нарисованные прямоугольники образуют квадрат, или «NO» в противном случае. Примеры  
входные данные  
5  
0 0 2 3  
0 3 3 5  
2 0 5 2  
3 2 5 5  
2 2 3 3  
выходные данные  
YES

## Задача №5

Вам задана таблица a, состоящая из n строк, пронумерованных от 1 до n. В i-ой строке таблицы a содержится ci клеточек, при этом для всех i (1 < i ≤ n) выполняется ci ≤ ci - 1.

Обозначим за s общее количество клеточек таблицы a, то есть . Известно, что в каждой клеточке таблицы a записано единственное целое число от 1 до s, при этом все записанные числа различны.

Пусть клеточки i-той строки таблицы a пронумерованы от 1 до ci, тогда обозначим число, записанное в j-той клеточке i-той строки, за ai, j. Вам необходимо посредством нескольких операций обмена переупорядочить числа в таблице так, чтобы выполнялись следующие условия:

для всех i, j (1 < i ≤ n; 1 ≤ j ≤ ci) выполняется ai, j > ai - 1, j; для всех i, j (1 ≤ i ≤ n; 1 < j ≤ ci) выполняется ai, j > ai, j - 1. За одну операцию обмена разрешается выбрать две различных клеточки таблицы и поменять записанные в них числа местами, то есть число, которое до применения операции было записано в первой из выбранных клеток, после применения операции будет записано во второй. Аналогично, число, которое до применения операции было записано во второй из выбранных клеток, после применения операции будет записано в первой.

Переупорядочите числа требуемым способом. Учтите, что Вам разрешается выполнить любое количество операций не превосходящее s. Минимизировать количество операций не требуется.

Входные данные В первой строке записано единственное целое число n (1 ≤ n ≤ 50), обозначающее количество строк в таблице. Во второй строке через пробел записаны n целых чисел ci (1 ≤ ci ≤ 50; ci ≤ ci - 1) — количества клеточек в соответствующих строках.

В следующих n строках задана таблица а. В i-той из них через пробел записаны ci целых чисел: j-тое число в этой строке обозначает ai, j.

Гарантируется, что все заданные числа ai, j положительны и не превосходят s. Гарантируется, что все ai, j различны.

Выходные данные В первой строке выведите единственное целое число m (0 ≤ m ≤ s), обозначающее количество произведенных операций обмена.

В следующих m строках выведите описание этих операций обмена. В i-той из них выведите через пробел четыре целых числа xi, yi, pi, qi (1 ≤ xi, pi ≤ n; 1 ≤ yi ≤ cxi; 1 ≤ qi ≤ cpi). Выведенные числа обозначают операцию обмена содержимого клеток axi, yi и api, qi. Обратите внимание, что операцией обмена можно менять содержимое различных клеток таблицы. Выводите обмены в том порядке, в котором они должны производиться.

Примеры  
входные данные  
3  
3 2 1  
4 3 5  
6 1  
2  
выходные данные  
2  
1 1 2 2  
2 1 3 1

## Задача №6

Петя заметил, что когда он набирает текст на клавиатуре, у него часто нажимаются лишние клавиши и в словах возникают лишние буквы. Конечно же, система проверки правописания подчеркивает ему эти слова, ему приходится кликать на слово и выбирать правильный вариант. Пете надоело исправлять свои ошибки вручную, поэтому он решил реализовать функцию, которая сама будет вносить исправления. Петя начал с разбора наиболее часто встречающегося у него случая, когда из слова достаточно удалить одну букву, чтобы оно совпало с некоторым словом из словаря. Итак, Петя столкнулся с такой подзадачей: дано введенное слово и слово из словаря, нужно удалить из первого слова одну букву, чтобы получилось второе. И тогда перед Петей встал весьма нетривиальный вопрос: какую букву удалять?

Входные данные Входные данные содержат две строки, состоящие из строчных латинских букв. Длина строк от 1 до 106 символов включительно, первая строка содержит ровно на 1 символ больше, чем вторая.

Выходные данные В первой строке выведите количество позиций символов в первой строке, при удалении каждого из которых получается вторая строка. Во второй строке через пробел выведите сами позиции в порядке возрастания. Позиции нумеруются с 1. Если из первой строки невозможно получить вторую путем удаления одного символа, выведите одно число 0.

Примеры

входные данные

abdrakadabra

abrakadabra

выходные данные

1

3

## Задача №7

До войны Вася был бухгалтером, он один из немногих, кто умеет пользоваться компьютером, и поэтому программистом назначили именно его.

Все мы знаем, что в программах часто приходится хранить наборы целых чисел. Например, ребро в задаче про взвешенный ориентированный граф может характеризоваться тремя числами: номер вершины-начала, номер вершины-конца и вес ребра. Вот и Вася столкнулся с такой проблемой, пытаясь представить в своей программе характеристки недавно разработанного робота.

Вася — не программист, поэтому он спросил у своего друга Гены, как удобнее хранить n целых чисел. Раньше Гена писал на языке X-- и поэтому умеет пользоваться только типами из этого языка. Определим понятие типа в языке X--:

Во-первых, типом является строка «int». Во-вторых, типом является строка начинающаяся с «pair», после которой в треугольных скобках через запятую указаны ровно два других типа языка X--. В этой записи нет пробелов. Никакие другие строки типом не являются. Более формально: type := int | pair<type,type>. Например, для ребер графа Гена использует тип pair<int,pair<int,int>>.

Гена с радостью помог Васе, он продиктовал ему тип языка X--, который хранит n целых чисел. К сожалению, Гена торопился и не произносил знаков препинания. Теперь Гена уже ушел, а Вася все никак не может расставить знаки препинания, чтобы получить тип языка X--.

Помогите Васе, расставьте знаки препинания так, чтобы получился корректный тип языка X--, или сообщите, что сделать это невозможно.

Входные данные В первой строке записано единственное целое число n (1 ≤ n ≤ 105) — количество чисел, которое хранит тип продиктованный Геной.

Во второй строке через пробел записаны слова, сказанные Геной. Каждое из них — «pair» или «int» (без кавычек).

Гарантируется, что общее количество слов не превосходит 105 и что среди произнесенных слов будет ровно n слов «int».

Выходные данные Если можно расставить знаки препинания так, чтобы получился корректный тип языка X--, выведите единственную строку обозначающую полученный тип. Иначе выведите «Error occurred» (без кавычек). Запись типа не должна содержать внутри себя лишних пробелов и других символов.

Гарантируется, что если такой тип существует, то он единственен.

Обратите внимание, что нужно вывести тип, произнесенный Геной (если такой существует), а не любой, в котором помещается n значений.

Примеры

входные данные

3

pair pair int int int

выходные данные

pair<pair<int,int>,int>

входные данные

1

pair int

выходные данные

Error occurred

## Задача №8

В Берляндском государственном университете теперь можно учиться онлайн! Для получения диплома среди всего многообразия онлайн-курсов Поликарпу необходимо пройти k главных онлайн-курса его специальности. Всего для прохождения доступны n курсов.

Ситуация осложняется тем, что онлайн-курсы зависят друг от друга и для каждого курса есть список тех, которые необходимо обязательно пройти до начала прохождения этого онлайн-курса (список может быть пустым, что означает отсутствие таких ограничений).

Помогите Поликарпу пройти наименьшее количество курсов суммарно, чтобы получить cпециальность (то есть пройти все главные курсы и необходимые для их прохождения курсы). Напишите программу, которая выводит порядок прохождения курсов.

Поликарп проходит курсы последовательно, то есть он может приступать к прохождению следующего только после окончания предыдущего. Каждый из курсов можно проходить не более одного раза.

Входные данные В первой строке записаны n и k (1 ≤ k ≤ n ≤ 105) — количество онлайн-курсов и количество главных курсов специальности Поликарпа.

Вторая строка содержит k различных целых чисел от 1 до n — номера главных онлайн-курсов специальности Поликарпа.

Далее следует n строк, каждая из которых описывает очередной курс: i-я из них соответствует курсу i. Каждая строка начинается с целого числа ti (0 ≤ ti ≤ n - 1) — количества курсов, от которых зависит i-й. Далее следует последовательность из ti различных целых чисел от 1 до n — номера курсов в произвольном порядке, от которых зависит i-й. Гарантируется, что никакой курс не может зависеть сам от себя.

Гарантируется, что сумма по всем значениям ti не превосходит 105.

Выходные данные Выведите -1, если не существует способа получения специальности.

В противном случае в первую строку выведите целое число m — минимальное количество онлайн-курсов, которое необходимо пройти для получения специальности. Во вторую строку выведите m различных целых чисел — номера курсов, которые надо пройти, в хронологическом порядке их прохождения. Если ответов несколько разрешается вывести любой из них.

Примеры

входные данные

6 2

5 3

0

0

0

2 2 1

1 4

1 5

выходные данные

5

1 2 3 4 5

## Задача №9

Из пункта A в пункт B со скоростью v м/с едет машина. Действие происходит на оси Ox. На расстоянии d метров, считая от A, находится светофор. Начиная с момента времени 0, первые g секунд на светофоре говорит зеленый свет, следующие r секунд на нем говорит красный свет, затем снова g секунд зеленый и так далее.

Машина может мгновенно разгоняться от 0 до v и, наоборот, может мгновенно тормозить от v до 0. Считается, что она проезжает на светофоре на зеленый свет мгновенно. Если машина подъезжает к светофору в момент, когда на нем только загорелся красный, она не успевает проехать. Но если она подъезжает в момент, когда только загорелся зеленый, она может проехать. Машина выезжает из пункта A в момент времени 0.

За какое наименьшее время машина сможет доехать из пункта A до пункта B, не нарушая при этом правил дорожного движения?

Входные данные В первой строке записаны целые числа l, d, v, g, r (1 ≤ l, d, v, g, r ≤ 1000, d < l) — соответственно расстояние между A и B (в метрах), расстояние от A до светофора, скорость машины, время зеленого света, время красного света.

Выходные данные Выведите одно число — за какое наименьшее время машина сможет приехать из пункта A в пункт B. Ваш ответ должен иметь относительную или абсолютную погрешность меньше чем 10 - 6.

Примеры

входные данные

2 1 3 4 5

выходные данные

0.66666667

## Задача №10

Петя любит счастливые числа. Всем известно, что счастливыми являются положительные целые числа, в десятичной записи которых содержатся только счастливые цифры 4 и 7. Например, числа 47, 744, 4 являются счастливыми, а 5, 17, 467 — не являются.

У Пети есть число из n цифр без лидирующих нулей. Он представил его в виде массива цифр d (нумерация с 1, начиная от старших цифр). Петя хочет k раз выполнить следующую операцию: найти минимальное x (1 ≤ x < n) такое, что dx = 4 и dx + 1 = 7, если x нечетное, то присвоить dx = dx + 1 = 4, иначе присвоить dx = dx + 1 = 7. Учтите, что если x не нашлось, то операция засчитывается как выполненная, и массив не меняется вообще.

Вам дано исходное число (в виде массива цифр) и число k. Помогите Пете узнать результат выполнения k операций.

Входные данные В первой строке задано два целых числа n и k (1 ≤ n ≤ 105, 0 ≤ k ≤ 109) — количество цифр числа и количество сделанных операций. Во второй строке задано n цифр без пробелов — массив цифр d, начиная с d1. Гарантируется, что первая цифра числа не равна нулю.

Выходные данные В единственной строке выведите без пробелов результат — число после выполнения k операций.

Примеры

входные данные

7 4

4727447

выходные данные

4427477

# Выполнение

## Листинг кода задачи №1

from math import inf   
*def* validate\_input(n, low = -inf, high = inf):  
 if not *hasattr*(n, '\_\_iter\_\_'):  
 n = [n]  
 for i in n:  
 if not (low <= *int*(i) <= high):  
 raise *Exception*('Ошибка, значение {} не находится в диапазоне от {} до {}'.format(i, low, high))  
*def* validate\_inputs(restrictions:*list*):  
 for r in restrictions:  
 validate\_input(\*r)

*def* mixes(n, a):  
 MaxN = 100000  
 b = [0] \* n  
 c = [0] \* (MaxN + 1)  
 for i in *range* (n):  
 c[a[i]] += 1  
 b[i] = c[a[i]]  
 bad = *False*  
 num = c[1]  
 for i in *range*(1, MaxN):  
 if c[i + 1] > c[i]:  
 bad = *True*  
 num = *max* (num, c[i + 1])  
 if bad:  
 return -1  
 else:  
 return [num, b]  
  
n = *int*(*input*())  
validate\_input(n, 1, 105)  
a = [*int*(x) for x in *input*().split ()]  
validate\_input(a, 1, 105)  
  
res = mixes(n, a)  
*print*(res[0])  
*print*(\*res[1] if res[0] != -1 else '')

## Листинг кода задачи №2

*def* olymps(params):  
 s = [0] \* 466  
 y = [0, 100, 131, 159, 190, 220, 251, 281, 312, 343, 373, 404, 434]  
 for i in params:  
 m, d, p, t = i  
 x = y[m] + d  
 s[x] -= p  
 s[x - t] += p  
 for i in *range*(465):  
 s[i + 1] += s[i]  
 return s  
  
n = *int*(*input*())  
validate\_input(n, 1, 100)  
params = []  
for i in *range*(n):  
 m, d, p, t = [*int*(item) for item in *input*().split()]  
 validate\_inputs([[m, 1, 12], [d, 1], [p, 1], [t, -inf, 100]])  
 params.append([m, d, p, t])  
  
*max*(olymps(params))

## Листинг кода задачи №3

*def* to\_decimal(num: *int*, foun: *int*):  
 num = *str*(num)  
 res = 0  
 for i in *range*(*len*(num)):  
 res += *int*(num[i]) \* foun\*\*(*len*(num) - 1 - i)  
  
 return *int*(res)  
  
*def* from\_decimal(num: *int*, foun: *int*):  
 res = ""  
 while(num > 0):  
 mod = num % foun  
 res = *str*(mod) + res  
 num = num // foun  
 return res  
  
*def* longest\_num(nums):  
 digits = "".join(nums)  
 foun = *int*(*max*(digits)) + 1  
 sum\_dec = *sum*([to\_decimal(*int*(num), foun) for num in nums])  
 return *len*(*str*(from\_decimal(sum\_dec, foun)))  
   
nums = [num for num in (*input*()).split()]  
validate\_input(*list*(*map*(*int*, nums)), 1, 1000)  
longest\_num(nums)

## Листинг кода задачи №4

import math  
  
*def* can\_fit(coords):  
 minX, maxX, minY, maxY = [*None*] \* 4  
 s = 0  
 for i in coords:  
 x1, y1, x2, y2 = i  
 if (minX == *None* or minX > x1):  
 minX = x1  
 if (maxX == *None* or maxX < x2):  
 maxX = x2  
  
 if (minY == *None* or minY > y1):  
 minY = y1  
 if (maxY == *None* or maxY < y2):  
 maxY = y2  
  
 s += (x2 - x1) \* (y2 - y1)  
  
 a = *abs*(maxX - minX)  
 b = *abs*(maxY - minY)  
 return "YES" if a == b == math.sqrt(s) else 'NO'  
  
n = *int*(*input*())  
validate\_input(n, 1, 5)  
coords = []  
  
for *\_* in *range*(n):  
 x1, y1, x2, y2 = [*int*(coord) for coord in *input*().split()]  
 validate\_inputs([[x1, 0, x2-1], [x2, x1+1, 31400], [y1, 0, y2-1], [y2, y1+1, 31400]])  
 coords.append([x1, y1, x2, y2])  
   
can\_fit(coords)

## Листинг кода задачи №5

*def* sort\_pyr(arr, col, n):  
 mm = *sum*(col)  
 ans = [(0,0)]\*(mm+1)  
 res = []  
  
 for i in *range*(n):  
 for j in *range*(col[i]):  
 ans[arr[i][j]]=(i,j)  
  
 b = 0  
 for i in *range*(n):  
 for j in *range*(col[i]):  
 b += 1  
 if arr[i][j] != b:  
 res.append([i, j, ans[b][0], ans[b][1]])  
 arr[ans[b][0]][ans[b][1]] = arr[i][j]  
 ans[arr[i][j]] = ans[b]  
 return res  
  
n = *int*(*input*())  
validate\_input(n, 1, 50)  
  
col = *list*(*map*(*int*,*input*().split()))  
validate\_input(col, 1, 50)  
  
arr = [*list*(*map*(*int*, *input*().split())) for *\_* in *range*(n)]  
  
res = sort\_pyr(arr, col, n)  
  
*print*(*len*(res))   
for i in res:  
 for j in i:  
 *print*(j+1,end=' ')  
 *print*()

## Листинг кода задачи №6

*def* t\_nine(wrong, right):  
 errors = []  
 for i in *range*(*len*(wrong)):  
 if wrong[:i] + wrong[i+1:] == right:  
 errors.append(i+1)  
 i += 1  
 return errors  
  
wrong = *input*()  
right = *input*()  
  
if not (1 <= *len*(right) <= *len*(wrong) <= 106 and *len*(wrong) - *len*(right) == 1):  
 raise *Exception*('Введена строка неправильной длины')  
  
errors = t\_nine(wrong, right)  
*print*(*len*(errors))  
for err in errors:  
 *print*(err, end = ' ')

## Листинг кода задачи №7

*def* get\_type(oprts):  
 res = ''  
 stack = []  
 for oprt in oprts:  
 is\_int = oprt == 'int'  
 res += oprt  
 while is\_int and stack and stack[-1]:  
 res += '>'  
 stack.pop()  
 if stack is not *None*:   
 stack.pop()  
 stack.append(is\_int)  
 if (is\_int):  
 res += ','  
 else:  
 res += '<'  
 return res[:-1] if res.count('<') == res.count('>') else 'Error occurred'  
  
n = *input*()  
validate\_input(*int*(n), 1, 105)  
oprts = *input*().split()  
get\_type(oprts)

## Листинг кода задачи №8

*def* get\_course\_chain(draw, courArr):  
 results = []  
 prev = *None*  
 for i in draw:  
 tmp = []  
 if courArr[i-1] != []:  
 for c in courArr[i-1]:  
 if prev == c:  
 return -1  
 if c not in results and c not in tmp:  
 tmp.append(c)  
 draw.append(c)  
 if i not in results and i not in tmp:  
 tmp.append(i)  
 results = tmp + results  
 prev = i  
 return results  
  
n, k = *list*(*map*(*int*, *input*().split()))  
validate\_inputs([[k, 1, n], [n, k, 105]])  
  
draw = *list*(*map*(*int*, *input*().split()))  
validate\_input(draw, 1, n)  
courArr = []  
  
for *\_* in *range*(n):  
 item = *list*(*map*(*int*, *input*().split()))  
 validate\_inputs([[item[0], 0, n-1], [item[1:], 1, n]])  
 courArr.append(item[1:])  
  
res = get\_course\_chain(draw, courArr)  
if res != -1:  
 *print*(*len*(res))  
 *print*(\*res)  
else:  
 *print*(res)

## Листинг кода задачи №9

*def* min\_time(l,d,v,g,r):  
 t1 = d/v  
 tmp = t1  
 dif = 0  
  
 is\_green = *True*  
 while *True*:  
 if (is\_green and tmp > g):  
 tmp -= g  
 is\_green = *False*  
 elif (tmp > r):  
 tmp -= r  
 is\_green = *True*  
 else:  
 break  
  
 if (is\_green):  
 dif = 0  
 elif (tmp != 0):  
 dif = r - tmp  
  
 t2 = (l - d) / v  
 res = t1 + t2 + dif  
 return res  
  
l,d,v,g,r = [*int*(p) for p in *input*().split()]  
validate\_inputs([[[l, d, v, g, r], 1, 1000], [d, 1, l-1]])  
min\_time(l,d,v,g,r)

## Листинг кода задачи №10

*def* happy\_nums(d, k):  
 for *\_* in *range*(k):  
 for i in *range*(*len*(d)-1):  
 if d[i] == 4 and d[i+1] == 7:  
 if (i+1) % 2 == 1:  
 d[i] = 4  
 d[i+1] = 4  
 else:  
 d[i] = 7  
 d[i+1] = 7  
 break  
 *print*("".join(*str*(x) for x in d))  
n, k = [*int*(i) for i in *input*().split()]  
validate\_inputs([[n, 1, 105], [k, 0, 109]])  
d = [*int*(i) for i in *list*(*input*())]  
if *len*(d) != n:  
 raise *Exception*('Строка неверной длины')  
happy\_nums(d, k)

# Вывод

В ходе проделанной курсовой работы я выполнил 10 алгоритмических задач на языке Python.