Backend-разработка на Java. Весна 2024

Оглавление

[**Вступительный контест Backend-разработка на Java, весна 2024** 2](#_Toc155807158)

[**1 задание** 3](#_Toc155807159)

[**2 задание** 4](#_Toc155807160)

[**3 задание** 5](#_Toc155807161)

[**4 задание** 7](#_Toc155807162)

[**5 задание** 9](#_Toc155807163)

[**Вступительный тест Backend-разработка на Java, весна 2024** 11](#_Toc155807164)

# **Вступительный контест Backend-разработка на Java, весна 2024**

Вступительный контест, состоящий из 5 задач. Задачи можно решать на любом языке программирования. Экзамен ограничен по времени — вам дается 4 часа. У вас есть только одна попытка написать контест.

# **1 задание**

**Ограничение времени**

1 секунда

**Ограничение памяти**

512 МБ

Львенок Максим использует в качестве паролей только последовательности из цифр.

Все пароли Максим хранит в одном текстовом файле в формате «code[DIGITS]», где вместо [DIGITS] подставляются цифры очередного пароля. Кроме записей паролей, в этом файле содержатся другие строчные латинские буквы, которые должны помогать Льву вспоминать, от чего тот или иной пароль.

Однажды Максим решил разобраться в своих паролях и распечатать свой файл. Он не хочет печатать этот файл в явном виде с паролями, так как это не безопасно.

Помогите Львенку и напишите программу, которая заменит все вхождения паролей в файле (вместе со словом code) на «???» (три символа вопроса).

**Формат входных данных**

В первой и единственной строке входных данных дано содержимое текстового файла. Текстовый файл содержит только строчные латинские буквы и цифры. Гарантируется, что текстовый файл содержит не больше ﻿105﻿ символов.

**Формат выходных данных**

Выведите содержимое файла с паролями, в котором вместо каждого пароля будут выведены три вопросительных знака.

Примеры данных

Пример 1

Ввод

a1b2c3

Вывод

a1b2c3

Пример 2

Ввод

code12345

Вывод

???

Пример 3

Ввод

aaacodebcode1code2code3bbb

Вывод

aaacodeb?????????bbb

Пример 4

Ввод

code999andcodec1andcocode888

Вывод

???andcodec1andco???

# **2 задание**

**Ограничение времени**

1 секунда

**Ограничение памяти**

512 МБ

Заяц Миша очень любит читать, поэтому на новый год ему подарили ﻿*n*﻿ книг.

Михаил хочет сложить компактно все подаренные книги в несколько стопок. Заяц считает, что книги сложены в стопки красиво, если в каждой стопке все книги имеют одинаковую высоту.

Миша хочет сложить книги в минимальное число стопок так, чтобы книги были сложены красиво. Помогите ему и найдите, как нужно сложить книги в стопки.

**Формат входных данных**

В первой строке входных данных дано целое число ﻿*n*﻿ — число книг, подаренных Зайцу(﻿1 ≤ n ≤ 105 1 ≤ *n* ≤ 105). Во второй строке дано ﻿*n*﻿ целых чисел ﻿*x*1​,…,*xn*​﻿ — высоты книг(﻿1≤xi≤ 105 1 ≤ *xi*​ ≤ 105﻿).

**Формат выходных данных**

В первой строке вывода напечатайте одно число ﻿*k*﻿ — минимальное число стопок книг. Во второй строке выведите высоты стопок в порядке не убывания.

Примеры данных

Пример 1

Ввод

2

1 2

Вывод

2

1 1

Пример 2

Ввод

4

1 2 2 3

Вывод

3

1 1 2

Пример 3

Ввод

5

5 4 4 5 5

Вывод

2

2 3

# **3 задание**

**Ограничение времени**

2 секунды

**Ограничение памяти**

512 МБ

Костя ответственно относится к подаркам для своих друзей. В этот раз он решил сделать ﻿*n*﻿ подарков своими руками для своих лучших друзей.

Для изготовления подарков он заказал материалы, причём материал для подарка ﻿*i*﻿ придёт в день ﻿*di*​﻿. Также Костя знает, что на изготовление подарка ﻿*i*﻿ нужно потратить ﻿*ci*​﻿ дней. Костя трудолюбивый, поэтому в один день может готовить любое число подарков (даже все).

Готовые подарки Костя планирует отправить почтой. Чтобы подарок ﻿*i*﻿ успел дойти до нового года, его нужно отправить не позднее дня ﻿*si*​﻿. Кроме того, на почте действует ограничение: можно отправлять не больше одной посылки в один день.

Помогите Косте понять, успеет ли он приготовить и отправить все подарки так, чтобы они дошли до получателей вовремя.

**Формат входных данных**

В первой строке входных данных дано число ﻿*n*﻿ — число подарков, которое планирует приготовить Костя (﻿1≤*n*≤105﻿).

В следующих ﻿*n*﻿ строках дана информация про каждый из подарков. Описание подарка ﻿*i*﻿ состоит из трех целых положительных чисел ﻿*di*​﻿, ﻿*ci*​﻿ и ﻿*si*​﻿ — день, в который доставят материалы для изготовления подарка, время, необходимое на приготовление подарка, и день, до которого должен быть отправлен подарок, чтобы он успел дойти (﻿1≤*di*​,*ci*​,*si*​≤109﻿).

**Формат выходных данных**

Выведите «YES» (без кавычек), если Костя успеет приготовить и отправить все подарки вовремя, или «NO» (без кавычек) в противном случае.

**Замечание**

В первом тесте из условия Костя может отправить первый подарок в третий день, второй — в шестой день, а третий — в пятый.

Во втором тесте из условия Костя никак не может успеть отправить все подарки, так как он должен за 3 дня (второй, третий и четвертый) отправить четыре подарка.

В третьем тесте из Костя может отправить первый подарок в третий день, второй — в шестой день, третий — в четвертый день, а четвертый — в пятый.

Примеры данных



# **4 задание**

**Ограничение времени**

1 секунда

**Ограничение памяти**

512 МБ

Дракоша Катя хочет провести большую вечеринку. Она уже определилась с тем, сколько и каких угощений она приготовит для гостей, однако не определилась, кого звать.

Про каждого друга-дракона Катя знает две характеристики ﻿*ai*​﻿ и ﻿*bi*​﻿ — интересность и прожорливость. Некоторые драконы могут дружить друг с другом. Также считается, что драконы ﻿*i*﻿ и ﻿*j*﻿ знакомы, если они дружат или дружат через одного или нескольких общих друзей. Более формально, дракон *i*﻿ знаком с драконом ﻿*j*﻿, если существует последовательность драконов ﻿*i*=*c*1​,*c*2​,…,*ck*​=*j*﻿ такая, что дракон ﻿*ct*−1​﻿ дружит с драконом ﻿*ct*​﻿.

Катя не может позвать дракона ﻿*i*﻿, если не позовет дракона ﻿*j*﻿, который знаком с драконом ﻿*i*﻿. Если она так сделает, то дракон ﻿*j*﻿ узнает о вечеринке и обидится на Катю.

Катя хочет позвать такое множество друзей, чтобы суммарная прожорливость всех гостей не превышала ﻿*g*﻿, а интересность приглашенных драконов была максимальной. Помогите Кате завершить приготовления к вечеринке и определиться со списком гостей.

**Формат входных данных**

Первая строка входных данных содержит три целых числа ﻿*n*﻿, ﻿*m*﻿ и ﻿*g*﻿﻿*w*﻿ — число друзей Кати, число пар драконов, которые дружат друг с другом, и максимальная суммарная прожорливость друзей, на которую хватит угощений, заготовленных Катей (﻿1≤*n*≤103,0≤*m*≤105,1≤*g*≤103﻿).

Вторая строка входных данных содержит ﻿*n*﻿ целых чисел ﻿1,2,…,*a*1​,*a*2​,…,*an*​﻿ (﻿1≤*ai*​ ≤106﻿) — интересность драконов-друзей Кати.

Третья строка содержит n целых чисел ﻿*b*1​,*b*2​,…,*bn*​﻿ (1≤*bi*​ ≤103﻿) — прожорливость драконов-друзей Кати.

В следующих ﻿*m*﻿ строках дано описание пар друзей-драконов, которые дружат друг с другом напрямую. Каждая пара друзей задается номерами драконов ﻿*ui*​﻿ и ﻿*vi*​﻿ (﻿1≤*ui*​ , *vi*​ ≤*n*,*ui*​ !=*vi*​﻿). При этом все пары номеров друзей (﻿*ui*​, *vi*​﻿) различны, а отношение дракон ﻿*ui*​﻿ дружит с драконом ﻿*vi*​﻿ является симметричным.

**Формат выходных данных**

В единственной строке выходного файла выведите наибольшую возможную суммарную интересность драконов, которых Екатерина может позвать на вечеринку, так чтобы соблюсти все условия.

Примеры данных



# **5 задание**

Ограничение времени

4 секунды

Ограничение памяти

512 МБ

Единорог Тимофей отправился в путешествие по волшебному лабиринту.

Лабиринт состоит из ﻿*n*﻿ комнат, которые соединены переходами так, что между двумя любыми комнатами единственный путь.

Тимофей боится заблудиться в лабиринте, поэтому написал программу, которая помогает выбрать правильное направление при движении по лабиринту. Программа по номеру комнаты, в которой сейчас находится Тимофей, и комнате, в которую он хочет попасть, находит номер комнаты, в которую нужно перейти из той комнаты, в которой он находится, чтобы за минимальное число переходов добраться до той комнаты, в которую он хочет попасть.

Единорог потратил на написание программы 30 минут, а сможете ли вы написать такую же быстрее?

**Формат входных данных**

В первой строке входных данных дано одно целое число ﻿*n*﻿ — число комнат в лабиринте (﻿2≤*n*≤2⋅105﻿).

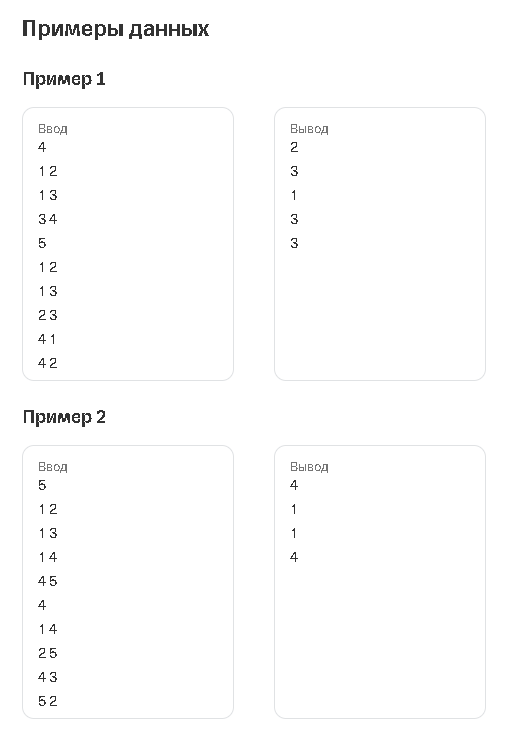
Далее дано ﻿*n*−1﻿ строк, в которых описаны переходы в лабиринте. Описание каждого перехода состоит из двух целых чисел — номеров узлов, которые он соединяет *ui*​﻿ и ﻿*vi*​﻿ (﻿1≤*ui*​ ,*vi*​ ≤*n*;*ui*​ != *vi*​﻿ ). Гарантируется, что между двумя любыми комнатами существует единственный путь.

В следующей строке дано одно целое число ﻿*q*﻿ — число вопросов о направлении, которые интересуют Тимофея (﻿1≤*q*≤2⋅105 ﻿).

В следующих ﻿*q*﻿ строках даны пары чисел ﻿*fromi*​﻿ , ﻿*toi*​﻿ — номера комнат, про направление движения между которыми хочет узнать Тимофей (1≤*fromi*​ ,*toi*​ ≤*n*;*fromi*​ != *toi*​﻿ ).

**Формат выходных данных**

Для каждого вопроса выведите номер комнаты, в которую нужно перейти из комнаты ﻿*fromi*​﻿, чтобы попасть в комнату ﻿*toi*​﻿ за минимальное число переходов.



# **Вступительный тест Backend-разработка на Java, весна 2024**

34 вопроса по основам Java. Экзамен ограничен по времени — на ответы на все вопросы вам дается **90 минут**. У вас есть только одна попытка пройти тест.