Демонстрационный вариант заключительного этапа

Всероссийской олимпиады школьников «Высшая проба»
по профилю «Промышленное программирование» для всех
классов 2024/2025 уч. г.

Блок «Теория»

Задача 1. Послание со звёзд

Учёные получили зашифрованное послание с Веги. По поляризации сигнала они поняли, что веганцы закодировали номер каждой буквы английского алфавита формулой $13\cdot(n+1)-n\%11$, записанной в троичной системе (% обозначает остаток от деления). Прочитайте сообщение, каждая буква которого записана сверху вниз.

```
122221121212221
010010110212220
010012011200120
122221120201 22
10000100201  01
     22 0     0
```

В ответе запишите **только строку** – расшифрованное сообщение прописными латинскими буквами.

Задача 2. Лабиринты

В Формиканской империи принята следующая система нумерации муравейников: есть номер сети родственных муравейников, который записывается в виде 24-разрядного двоичного числа (или в виде трех десятичных чисел через звездочку) и есть номер каждого отдельного муравейника в этой сети в таком же формате. У всех муравейников одной сети одинаковы первые части и уникальны вторые. Разделить общую и уникальную часть можно с помощью маски: тоже 24-разрядного двоичного числа, в котором сначала стоят единицы, а с некоторого места нули. При поразрядной конъюнкции номера муравейника и маски получится номер сети родственных муравейников.

Для сокращенного обозначения номера сети используется обозначение через “/” количества бит в маске, содержащих единицы.

Например, для муравейника с номером 176*23*13/16 маска такая: 255*255*0.

В некоторой родственной сети есть муравейник с номером 13*28*82/17. Внутри этой сети нужно создать подсеть с минимальным количеством адресов, достаточным для нумерации 13 муравейников.

Запишите адрес такой подсети и маски в таком же формате (например, 176*23*13/16). Если таких адресов несколько, то адрес с наибольшим значением.

Блок «Алгоритмы»

Задача 1. Робот-чертёжник

Робот-чертёжник закрашивает полосу из 10^100 клеток по следующей программе: первые K клеток он закрашивает красным цветом, следующие K клеток — синим цветом, следующие K клеток — снова красным, следующие K — снова синим, и так далее.

Когда робот раскрасил полосу, из неё вырезали фрагмент, состоящий из W идущих подряд клеток. Определите, сколько красных клеток может оказаться в этом фрагменте.

Формат ввода

Первая строка содержит целое число K (1 <= K <= 10^9) — количество подряд идущих клеток, окрашиваемых роботом в красный или синий цвет.

Вторая строка содержит целое число W (2 * K <= W <= 2 * 10^9) — количество клеток в вырезанном фрагменте.

Формат вывода

В первой строке выведите одно целое число — минимально возможное количество красных клеток в вырезанном фрагменте.

Во второй строке выведите одно целое число — максимально возможное количество красных клеток в вырезанном фрагменте.

Пример 1

Ввод
1
2

Вывод
1
1

Пример 2

Ввод
5
23

Вывод

10

13

Примечания

Если ваше решение даёт верный ответ для W, делящемся нацело на 2 * K, вы получите не менее 30% баллов за эту задачу.

Задача 2. Блины

Вы открыли пекарню и продаёте блины. Как известно, блин состоит из теста и начинки.

В вашей пекарне есть N видов теста, тесто i-го вида стоит A_i рублей. Также в вашей пекарне есть N видов начинки, начинка i-го вида стоит B_i рублей.

Клиент может выбрать любое тесто и любую начинку. Так как вам нужно зарабатывать, стоимость итогового блина вы определяете как произведение стоимости теста и стоимости начинки (например, если клиент выбрал тесто стоимости 10 и начинку стоимости 12, стоимость блина составит 10 * 12 = 120).

На прошлой неделе к вам пришли K клиентов. Вы не помните, что именно заказывал каждый из них, но запомнили, что все клиенты заказали разные блины (то есть у любых двух заказов отличались либо тесто, либо начинка, либо и то, и другое). Кроме того, вы помните, что стоимость каждого следующего заказа была не меньше, чем у предыдущего.

Вы пытаетесь вспомнить, сколько стоил заказ самого первого клиента. Найдите его максимально возможную стоимость.

Формат ввода

Первая строка содержит целые числа N и K (1 <= N <= 10^5, 1 <= K <= N^2) — соответственно количество видов теста и начинки и количество клиентов.

Вторая строка содержит N целых чисел A_i (1 <= A_i <= 10^9) — стоимости каждого из видов теста.

Третья строка содержит N целых чисел B_i (1 <= B_i <= 10^9) — стоимости каждого из видов начинки.

Формат вывода

Выведите одно целое число — максимально возможную стоимость блина, заказанного самым первым клиентом.

Пример 1

Ввод

2 3
1 2
3 4

Вывод
4

Пример 2

Ввод
4 10
1 2 1 3
4 3 1 3

Вывод
3

Задача 3. Гармония

Вам нравятся числа, в которых количество чётных цифр совпадает с количеством нечётных цифр, например, 12, 6455 или 734054.

Подсчитайте, сколько чисел от L до R включительно обладают таким свойством.
Формат ввода

Первая строка содержит целое число L (1 <= L <= 10^18) — начало отрезка для поиска чисел.

Вторая строка содержит целое число R (L <= R <= 10^18) — конец отрезка для поиска чисел.

Формат вывода

Выведите одно целое число — количество чисел от L до R, у которых чётных и нечётных цифр поровну.

Пример 1

Ввод
10
20

Вывод
5

Пример 2

Ввод
1
1000

Вывод
45

Примечания

Если ваше решение даёт верный ответ для R <= 10^4, вы получите не менее 10% баллов за эту задачу.

Если ваше решение также даёт верный ответ для L = 10^X, R = 10^Y - 1, 0 <= X < Y <= 18, вы получите не менее 30% баллов за эту задачу.

Блок «Данные»

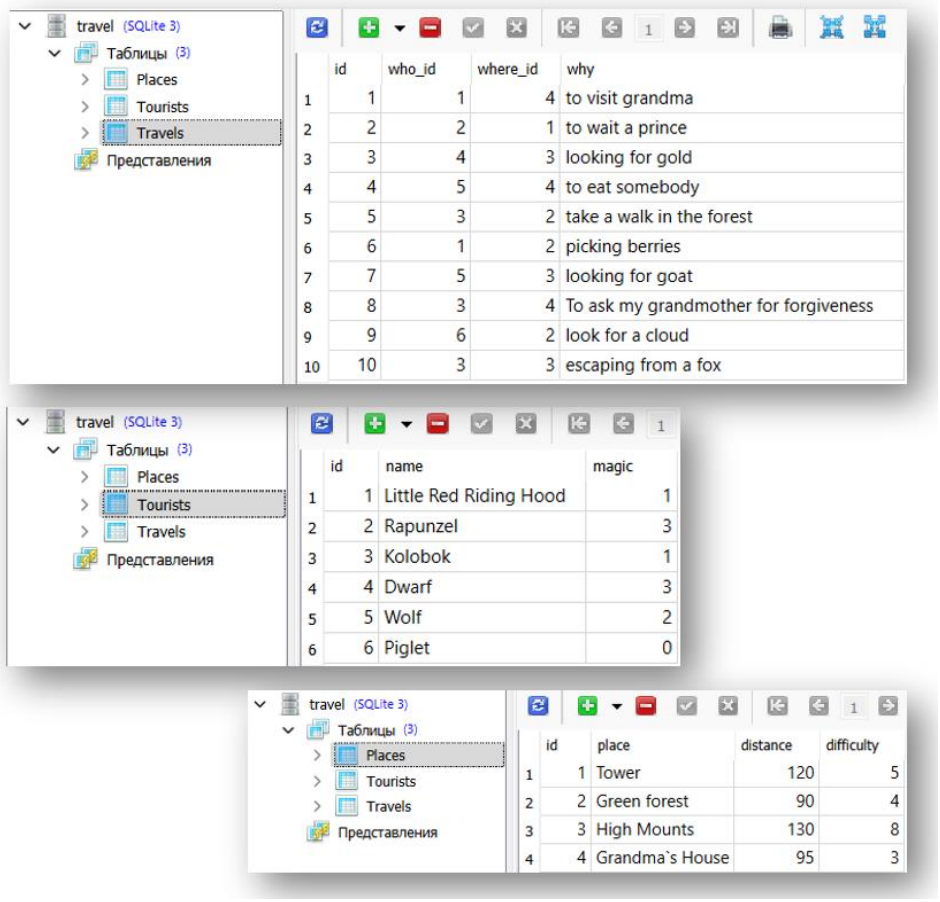
Задача 1. Все любят путешествовать

Все любят путешествовать! В базе данных **travel.db** записана информация о путешествиях в сказках в виде таблиц:

Travels (Путешествия)
id, who_id, where_id, why
id, кто, куда, зачем

Places (Место назначения)
id, place, distance, difficulty
id, место, дальность, трудность

Tourists (Путешественники)
id, name, magic
id, имя, магические способности
На рисунке показана структура БД и пример заполнения.



Напишите SQL-запрос, который вернёт имя путешественника, среднюю дальность его путешествий, среднюю трудность с группировкой по путешественнику и сортировкой по имени

в обратном алфавитном порядке для всех, кто обладает хоть какими-то магическими способностями (большими нуля).

Ваше решение должно содержать только скрипт, написанный на языке SQL (диалект SQLite), работающий с базой данных, аналогичной по структуре travel.db.

Пример базы данных travel.db для отладки решения можно скачать здесь.

Формат вывода

В ответе должно быть 3 столбца, названия столбцов по вашему усмотрению, порядок – как указано в задаче.

Примечания

Ваш запрос будет запущен на различных тестовых данных. Для базы данных travel.db из примера для отладки выводимый результат должен быть таким:

	name	avg_dist	avg_dif
1	Wolf	112.5	5.5
2	Rapunzel	120	5
3	Little Red Riding Hood	92.5	3.5
4	Kolobok	105	5
5	Dwarf	130	8

Задача 2. Подлинник

В доисторические времена муравьи и динозавры заключили пакт о взаимопомощи, в котором каждая цивилизация взяла на себя некоторые обязанности.

Однако по прошествии веков, когда возник спор, кто и что обещал делать, оказалось, что существует несколько версий этого важного документа.

Какой же из вариантов подлинный? Напишите программу, которая позволит это установить.

Формат ввода

Вводится имя файла *json* с ключами:
word – длина самого длинного слова (длина должна быть больше значения по ключу)
length – четность длины строки (длина строки должна иметь такую же четность)
consonants – количество не гласных букв (количество должно быть больше значения по ключу)
substr – подстрока (не должно быть такой подстроки)
upper – количество символов в верхнем регистре (должно быть больше значения по ключу)

Затем вводятся строки, каждая из которых – часть документа, для которой требуется установить подлинность.

Формат вывода

В файл **identity.csv** запишите данные по каждой строке с заголовками (разделители – точка с запятой):
i, word, length, val_word, val_length, val_cons, val_substr, val_upper, identity
индекс строки, длина самого длинного слова, длина строки, проверка по длинному слову, по четности, по не гласным, по подстроке, по регистру, общая подлинность

Проверка на подлинность по каждому признаку возвращает 0 или 1: например, проверка по длине самого длинного слова равна 1, если его длина больше значения по ключу “word” в словаре json. Проверка по подстроке возвращает 1, если подстрока, указанная в значении по ключу *“substr”*, отсутствует в строке. По остальным ключам проверка аналогичная.

Сообщение считается недостоверным (валидация равна 0), если сумма значений по полям валидации меньше 3.

Словом в этой задаче считается последовательность символов, ограниченная началом или концом строки или пробелом. Гласные – это *aeouiy*.

Пример

Ввод	Вывод
messages.json The balance between the two civilizations is extremely fragile, and any spark between them can ignite a fire. The unacceptably slow development of writing has become the biggest obstacle to scientific and cultural progress in the dinosaur world. We can say that the nascent intelligence is like a small light in an open field it can be extinguished by a light breeze from any direction. In such difficult times, wisdom is a sorrow, and unreason is a blessing. The simple redundancy of the collective mind cannot produce sublime thinking. Wisdom is like blades of grass sprouting with the onset of heat on a bare, glacier-stripped earth.	i;word;length;val_word;val_length;val_cons;val_substr;val_upper;identity 0;13;109;1;1;0;0;1;1 1;12;135;1;1;1;0;1;1 2;12;140;1;0;1;1;1;1 3;9;72;0;0;0;0;1;0 4;10;77;0;1;0;1;1;1 5;16;98;1;0;0;1;1;1

Примечания

Файл *messages.json* из примера содержит данные:

```
{
   "word": 10,
   "length": 1,
   "consonants": 60,
   "substr": "and",
   "upper": 1
}
```

В примере вывода показано содержимое файла.

Блок «Бэкенд»

Задача 1. Кулинарные изыски

А не приготовить ли нам чего-нибудь вкусненького?
Хорошо бы, а продукты для вкусненького у нас есть?
Сейчас узнаем.

В файле dishes.json, который доступен только в начале работы сервера, в формате JSON указан список доступных продуктов и список блюд, которые можно попытаться приготовить, с указанием необходимых продуктов, например, так:

```
{
  "available": ["tomato", "mozzarella"],
  "recipes": {
    "Salad": ["tomato", "mozzarella"],
    "Salad premium": ["tomato", "mozzarella", "salt"]
  }
}
```

Напишите сервер, который может определить, есть ли все необходимые продукты для приготовления блюда dish, выполнив запрос:
/stop/<disch>/ – вернёт название блюда в двойных кавычках и available через пробел, если блюдо можно приготовить с имеющимися продуктами; если нельзя, то вернёт название блюда (также в двойных кавычках) и stop.

Примеры запросов и ответов:

/stop/Salad/
"Salad" available

/stop/Salad premium/
"Salad premium" stop

Если не удалось определить, что это за блюдо, необходимо вернуть статус 404.

Задача 2. Коллекционер

У каждого предмета в коллекции есть свойства или характеристики, на основе которых их можно распределять по темам, разделам, подразделам… Например, цвет, размер, год выпуска, страна происхождения и т. д. Но самое ценное, когда удаётся собрать серию – набор предметов, которые отличаются только по одному какому-то свойству, а по всем остальным, которые определены хоть у одного из выбранных предметов, совпадают.

Напишите сервер, который обрабатывает запросы вида
/add/<object_id>/?color=white&size=small&... (количество ключей не известно заранее)

и после каждого запроса возвращает список id всех предметов (отсортированном в лексикографическом порядке, через запятую, без пробелов), которые с добавленным образуют серию.

Гарантируется, что id у всех предметов уникальны.

Пример

/add/1/?color=white&size=small
1
/add/2/?color=white&size=big
1,2
/add/3/?color=red&size=small
1,3
/add/4/?color=red&size=small&origin=china
4

Блок «Фронтенд»

Задача 1. Межвременная дефрагментация

Сотрудники отдела Темпоральной Обработки Информации нашли новый способ почти бесконечно увеличивать объём физических носителей данных. Оказывается, если выстроить непрерывный кварковый блок данных между антикварковым покрытием стенок носителя, то аннигилированные частицы одного типа попадают в межвременное пространство, к которому всё ещё можно получить доступ по флуктуациям в пространстве-времени от высвобождения фотонов. Для тестирования способа учёные придумали устройство, которое позволяет вручную дефрагментировать данные. Вам нужно создать визуальный интерфейс для этого устройства.

Правила игры простые:

- карта носителя данных может состоять из стенок, пустых полей, подвижного блока и зафиксированных блоков;
- подвижный блок может перемещаться на одно поле влево, вправо и вниз с помощью кнопок управления: arrow-left arrow-down arrow-right;
- подвижный блок может поворачиваться на 90 градусов (относительно центральной клетки) и менять цвет (тип кварка) между двумя возможными с помощью кнопок управления: arrow-up close;
- полный список возможных подвижных блоков (с учётом порядка при изменении):

```
`
[
    `
    o
    o
    o
    `,
    `
    q
    q
    q
    `,
    `
    ooo
    `,
    `
    qqq
    `
]
`;
```

- после движения вправо-влево и после замены подвижный блок дополнительно двигается вниз на одно поле;

- если движение вправо-влево или замена невозможны из-за близости стенок или зафиксированных блоков, то действие игнорируется (при этом дополнительное движение вниз всё равно происходит);
- когда подвижный блок при движении вниз упирается в стенку или зафиксированный блок, он фиксируется;
- после фиксации подвижного блока все полные горизонтальные линии зафиксированных блоков одного цвета исчезают;
- после фиксации подвижного блока (и исчезновения полных горизонтальных линий одного цвета, если таковые были) следующий блок появляется вверху карты и посредине по горизонтали (все карты и блоки нечётной ширины);
- если новому блоку негде появиться (из-за накопившихся зафиксированных блоков), то дефрагментация заканчивается — все зафиксированные блоки превращаются в стенки.

Формат ввода

Карта носителя доступна в поле window.map типа String.

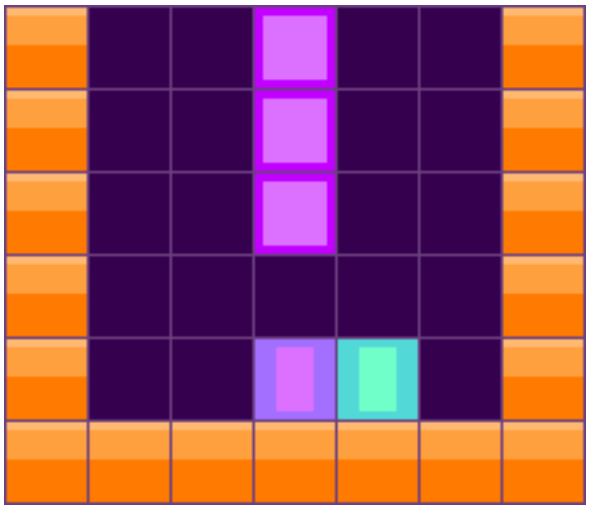
Специальные символы:

- `#` — стенка;
- `.` — пустое место;
- `o`, `q` — часть подвижного блока (два разных цвета);
- `O`, `Q` — зафиксированная часть блока (два разных цвета).

Каждая строка, содержащая хотя бы один специальный символ, является линией карты, каждый символ является полем карты. Любые другие символы не имеют значения и должны быть проигнорированы при построении карты.
Например,

```
window.map = `
    #..o..#
    #..o..#
    #..o..#
    #.....#
    #..OQ.#
    #######
`;
```

должно дать такой результат:



Примечания

Все карты в наших тестах имеют прямоугольную форму (нечётной ширины) и ограничены стенками по бокам и снизу.

Решение должно представлять из себя один HTML-файл, содержащий все нужные скрипты и стили.

После того, как интерфейс будет проинциализирован, надо вызвать глобальную функцию window.onGameReady(). Только после этого будет запущено автоматическое тестирование вашего решения. Если вызов функции не произойдёт в течение 2 минут, то задание считается невыполненным.

Контейнер с картой должен иметь CSS-класс map.

Кнопки управления должны реагировать на событие click и иметь следующие CSS-классы:

- влево — control_action_left;
- вниз — control_action_down;
- вправо — control_action_right;
- повернуть блок — control_action_switch-figure;
- поменять цвет блока — control_action_switch-color.

Значения цвета для разных элементов карты:

- задний фон: 35004d;
- декоративная сетка: 6c4080;
- стенка: ffb971, ff7a00, ffa03f;
- фигура первого цвета: c100ff, dd71ff;
- фигура второго цвета: 00ff73, 71ffc9.