B. Галактические Вести (15 баллов)

Космическая журналистика — только для настоящих межзвёздных акул пера. Редакция издания «Галактические Вести» пользуется самыми надёжными способами связи со своими косморепортёрами и даже придумала свой формат передачи данных.

Для отправки сообщения необходимо указать адрес получателя, добавить разделитель в виде символа двоеточия и ввести текст сообщения в фигурных скобках. Сообщение может содержать любой текст, а вот адрес задается согласно правилу: галактика/планетарная\_система/планета. Название галактики содержит от 2 до 8 заглавных латинских букв, дефис, а затем от 2 до 8 цифр. Планетарная система — непустая последовательность заглавных латинских букв и дефисов, в которой запрещено использовать два дефиса подряд, а также дефисы в начале и в конце последовательности. Планета — непустая последовательность заглавных латинских букв. Из-за религиозных взглядов главного редактора издания запрещено использовать подстроку UNDEFINED внутри названия планеты.

Издания-конкуренты не дремлют: постоянно норовят перехватить сообщения ваших репортёров и изменить их до неузнаваемости. Они почему-то всегда вставляют свои правки внутри специальных символов @, например @фейковая новость@. Вам точно известно, что символы всегда парные, никогда не вкладываются друг в друга, а используют их только конкуренты.

Ваша задача — разработать устройство, которое будет выхватывать из межзвёздного эфира сообщения ваших репортёров. Для этого нужно отсеять все сообщения, которые не соответствуют протоколу редакции, и, если в сообщении есть послания от конкурентов, обернуть каждое из них в теги <fake></fake> вместо символов @, чтобы их можно было передать в отдел фейковых новостей.

Формат ввода

[    
    "GALAXY-42/SYSTEM/PLANET:{}",    
    "GALAXY-42/SYSTEM/PLANET:{Message}",    
    "GALAXY-42/System/PLANET:{Message}",    
    "GALAXY-42/SYSTEM/PLANET{Message}",    
    "LONGGALAXY-42/SYSTEM/PLANET:{Message}",    
    "GALAXY-4/SYSTEM/PLANET:{Message}",    
    "GALAXY-4815162342/SYSTEM/PLANET:{Message}",    
    "GALAXY-42/THE-SOLAR-SYSTEM/PLANET:{Message}",    
    "GALAXY-42/-SYSTEM/PLANET:{Message}",    
    "GALAXY-42/SYSTEM1/PLANET:{Message}",    
    "GALAXY-42/SYS--TEM/PLANET:{Message}",    
    "GALAXY-42/ERROR/NIL:{Message}",    
    "GALAXY-42/ERROR/NULL:{Message}",    
    "GALAXY-42/ERROR/UNDEFINED:{Message}",    
    "GALAXY-42/TYPE/ISNOTUNDEFINED:{Message}",    
    "GALAXY-42/SYSTEM/PLANET:{Simple text... @null == undefined@}",    
    "GALAXY-42/SYSTEM/PLANET:{@typeof null@@typeof typeof null@}"    
]

Формат вывода

[    
    "GALAXY-42/SYSTEM/PLANET:{}",    
    "GALAXY-42/SYSTEM/PLANET:{Message}",    
    "GALAXY-42/THE-SOLAR-SYSTEM/PLANET:{Message}",    
    "GALAXY-42/ERROR/NIL:{Message}",    
    "GALAXY-42/ERROR/NULL:{Message}",    
    "GALAXY-42/SYSTEM/PLANET:{Simple text... <fake>null == undefined</fake>}",    
    "GALAXY-42/SYSTEM/PLANET:{<fake>typeof null</fake><fake>typeof typeof null</fake>}"    
]

Примечания

Файл с решением требуется оформить по шаблону:

module.exports = function (input) {    
    // ...    
    return result;    
}

Набрать здесьОтправить файл

Выбрать

# C. Проблемы с цветами (40 баллов)

На недавно колонизированной людьми планете RG189733b условия почти такие же, как на Земле, кроме одного обстоятельства — пыльца местных растений влияет на зрительную кору головного мозга человека так, что разные цвета воспринимаются с разной задержкой. Этот эффект чуть-чуть сводит с ума неподготовленных колонистов, поэтому нейробиологи написали программу для виртуальной реальности, которая имитирует действие пыльцы. Пока земляне летят к планете RG189733b, они каждый день проводят в виртуальной реальности по несколько часов, чтобы привыкнуть к будущим условиям жизни.

К сожалению, нейробиологи — не программисты, поэтому их программа работает очень медленно. Нужно оптимизировать программу, чтобы помочь колонистам в адаптации.

У вас есть исходный код функции, которая принимает на вход два аргумента:

* **data**— [UInt8ClampedArray](https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Global_Objects/Uint8ClampedArray) с битмапом кадра (каждые 4 байта - один пиксель в формате rgba),
* **api**— специальный API с методами:
  + **getDimensions**возвращает объект с размерами кадра,
  + **getTime**возвращает текущее время в миллисекундах.

Функция возвращает Promise с массивом типа UInt8ClampedArray пикселей картинки-результата.

Нужно ускорить эту функцию, не изменяя её сигнатуру и не сломав возвращаемый ею результат.

## Примечания

Если решение слишком медленное, то результатом теста будет TL (Time Limit) или неполный балл за тест.

В архиве по ссылке «скачать условие задачи» вы найдёте исходный код функции и страницу, которую можно использовать для отладки.

# D. Курсовая и закон Мёрфи (40 баллов)

|  |  |
| --- | --- |
| Ограничение времени | 1 секунда |
| Ограничение памяти | 64Mb |
| Ввод | стандартный ввод или input.json |
| Вывод | стандартный вывод или output.json |

Студент Артемий пренебрегает бэкапами и не пользуется системами контроля версий, потому что у него есть темпоральный излучатель, который позволяет доставать данные из прошлого. Но по закону Мёрфи прямо перед защитой курсовой работы по алгоритмам сортировки одногруппники нашли досадный баг в алгоритме Артемия, а излучатель он забыл дома. Исходные коды алгоритма Артемий посчитал бесполезным бэкапом, поэтому чинить придется обфусцированную против системы «Антиплагиат-2120» версию. Помогите Артемию получить допуск к сессии.

## Примечания

Чтобы тест-кейсы отработали правильно, нужно оставить yield в исходном алгоритме сортировки на своих местах.

[Скачать условие задачи](https://contest.yandex.ru/yacup_fe/contest/20977/download/D/)

E. Межвременная дефрагментация (60 баллов)

|  |  |
| --- | --- |
| Ограничение времени | 10 секунд |
| Ограничение памяти | 640Mb |
| Ввод | input.js |
| Вывод | output.png |

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.

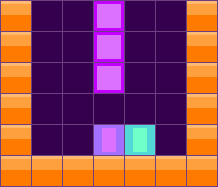
**Специальные символы:**

* # — 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.

Обратите внимание, ваше решение будет тестироваться в браузере **Google** **Chrome 69**.

# F. Самое ценное (60 баллов)

**Самое ценное**  
Инженеры космического корабля AVOS стали замечать, что на жидкие диски памяти корабля перестали помещаться новые подборки с сериалами. Чтобы расчистить диски для самого ценного, коллегиально было принято решение удалить неиспользуемый код из проекта модуля жизнеобеспечения корабля.  
**Неиспользуемым кодом**будем называть программный код, который существует в файловой системе в директории проекта, но никак не влияет на итоговый бандл.  
**Входная точка (entrypoint)**— исходный модуль в проекте, код из которого будет исполнен в одном из окружений.  
**Бандл (bundle)**— файл, содержащий программный код, являющийся результатом сборки. Код попадает в бандл, если существует зависимость между ним и входной точкой.

Вам нужно реализовать функцию, которая поможет инженерам найти неиспользуемые модули.

## Формат ввода

Декларативное описание документа выглядит следующим образом:

{    
    // абсолютный путь до дирректории проекта в файловой системе    
    absoluteRepoPath: "/var/www/projects/project1",    
    // список алиасов по путям из исходной системы сборки    
    aliases: {    
        "@": "/var/www/projects/project1/src",    
    },    
    // список всех входных точек приложения    
    entrypoints: [    
        "/var/www/projects/project1/src/pages/a.js",    
        "/var/www/projects/project1/src/pages/b.js",    
    ],    
    // информация о всех модулях данного проекта    
    modules: [    
        {    
            // относительный от корня путь    
            file: "./src/pages/a.js",    
            deps: [    
    // валидная для исходной системы сборки строка, описывающая путь до модуля    
    // гарантируется, что такой модуль существует и описан в данной секции    
                "/var/www/projects/project1/src/a.js",    
            ],    
        },    
        {    
            file: "./src/pages/b.js",    
            deps: [    
                "@/a.js",    
            ],    
        },    
        {    
            file: "./src/a.js",    
            deps: [],    
        },    
        {    
            file: "./src/f.js",    
            deps: [],    
        },    
        {    
            file: "./src/g.js",    
            deps: [],    
        },    
    ],    
}

## Формат вывода

Результатом работы алгоритма должен быть упорядоченный лексикографически массив неиспользуемых модулей, где каждый модуль описывается через абсолютный путь в файловой системе:

[    
    "/var/www/projects/project1/src/f.js",    
    "/var/www/projects/project1/src/g.js"    
]

В случае, если все модули используются, указывается пустой массив.

[]

## Примечания

Файл с решением требуется оформить по шаблону:

module.exports = function (input) {    
    // ...    
    return result;    
}