



Демонстрационный вариант отборочного этапа

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



ВЫСШАЯ ШКОЛА ЭКОНОМИКИ
НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ

Блок «Теория»	3
Задача 1	3
Задача 2	3
Задача 3	3
Задача 4	3
Блок «Алгоритмы»	5
Задача 1	5
Задача 2	5
Задача 3	6
Блок «Данные»	7
Задача 1	7
Задача 2	7
Задача 3	8
Блок «Бэкенд»	11
Задача 1	11
Задача 2	11
Задача 3	12
Блок «Фронтенд»	14
Задача 1	14
Задача 2	14
Задача 3	15

Блок «Теория»

Задача 1

Задумано число в 9-ричной системе счисления. Про него известно:

- произведение всех его цифр равно 21
- сумма его цифр, не превышающих 3, равна 9
- оно наибольшее из всех возможных

Запишите это число в 10-й системе счисления.

Задача 2

На экран транслируется прямоугольное изображение с соотношением сторон 16:9. Само изображение составлено из маленьких одинаковых квадратных плиток, на которых нарисованы символы из заданного набора из 262144 вариантов. Общее количество информации в одном кадре равно 81 Тб.

Определите количество плиток по горизонтальной (большей) стороне экрана.

Задача 3

В автоматическом пункте выдачи заказов ячейки арендуют несколько поставщиков. Для адресации ячеек используется 24-битный двоичный код, для удобства разделенный на три числа по 8 бит, переведенных в 10-ю систему. Числа записываются через дефис.

Каждому поставщику присвоен собственный номер, который занимает часть 24-битного адреса ячейки, остальная часть относится к собственному номеру ячейки. Чтобы из адреса ячейки получить номер поставщика, используется маска: также 24-битное число, в котором в старших разрядах стоят единицы, а начиная с некоторого разряда, – нули. Разделение происходит с помощью поразрядной конъюнкции адреса ячейки и маски, получается номер поставщика.

Для заданных адреса поставщика и адреса одной ячейки определите, какое наибольшее количество ячеек может быть отведено этому поставщику. Ячейки нумеруются с 1.

Адрес поставщика: 197-44-0

Адрес ячейки: 197-45-201

Задача 4

В цепных ядерных реакциях альтернативной вселенной, в которой существуют частицы, состоящие из трех видов альтернативных кварков – А, В и С – и глюона, обозначаемого О, выполняются следующие физические законы:

- реакция продолжается до тех пор, пока два глюона не окажутся рядом (ОО);
- если в цепочке оказалась подчастица ОА, то она превращается в ВВО;
- если в цепочке оказалась подчастица ОВ, то она превращается в ВОСА;

Промышленное программирование 9–11 классы ДЕМО

- если в цепочке оказалась подчастица ОС, то она превращается в ААОА.

Если после реакции распада частицы, состоящей из одного глюона в начале и одного глюона в конце, а между ними некоторого количества кварков, получилось 92 кварка А и 144 кварков В, то какое наименьшее количество кварков и глюонов могло входить в частицу до распада?

Блок «Алгоритмы»

Задача 1

Ремонт на даче

Ограничение времени: 1 секунда

Ограничение памяти: 256 Mb

Вы решили сделать ремонт на даче. Сейчас ваше внимание сосредоточено на одной конкретной комнате, где вы хотите покрасить все стены, а также пол и потолок в один цвет. Размеры комнаты: A метров в длину, B метров в ширину, C метров в высоту. Краска в сельском магазине продается только в банках по V литров ценой P рублей за банку. Один литр краски позволяет покрасить один квадратный метр поверхности.

Вычислите суммарную стоимость банок, которые вам придется приобрести.

Входные данные

В первой строке вводятся целые числа A, B, C — размеры комнаты ($1 \leq A, B, C \leq 100, 1 \leq A, B, C \leq 100$).

Во второй строке вводятся целые числа V, P — объем одной банки и ее цена ($1 \leq V, P \leq 100, 1 \leq V, P \leq 100$).

Выходные данные

Выведите одно число — суммарную стоимость банок краски, которые вам придется приобрести.

Задача 2

Иван и суперкратные числа

Ограничение времени: 2 секунды

Ограничение памяти: 256 Mb

Рассмотрим бесконечный ряд неотрицательных целых чисел A_i , кратных K . Например, для $K=3$ этот ряд начинается так:

$$A_0=0, A_1=3, A_2=6, A_3=9, A_4=12, A_5=15, A_6=18\dots$$

Иван называет число суперкратным K , если его номер в списке кратных чисел тоже делится на K . Например, число $A_6=18$ является суперкратным $K=3$, так как оно само делится на 3 и его номер 6 тоже делится на 3. Другими числами, суперкратными $K=3$, являются, например, 0 и 9.

Иван хочет найти максимальное число, суперкратное K и не превышающее N . Помогите ему это сделать.

Входные данные

Первая строка содержит целое число K ($1 \leq K \leq 2 \cdot 10^9$) — множитель для суперкратных чисел.

Промышленное программирование 9–11 классы ДЕМО

Вторая строка содержит целое число N ($K \leq N \leq 2 \cdot 10^9$) — верхнюю границу для поиска суперкратного числа.

Выходные данные

Выведите максимальное число, суперкратное K и не превышающее N .

Задача 3

Уникальный никнейм

Ограничение времени: 1 секунда

Ограничение памяти: 256 Mb

Егор прошел отбор на позицию ML-исследователя в Лаборатории Программной Инженерии. В первый рабочий день ему предстоит выбрать себе никнейм для корпоративного портала. Формально, никнейм — строка длины k , которая состоит из букв английского алфавита, причем первая буква никнейма всегда заглавная, а все остальные строчные.

Корпоративный портал требует создать уникальный никнейм, который отличается от всех n никнеймов, которые уже созданы в корпоративном портале. Но Егор хочет выделиться и решил придумать никнейм, который будет отличаться от никнеймов его коллег **в каждом символе**. Это значит, что каждый из k символов никнейма Егора должен отличаться от символов, которые использовали его коллеги на такой же по порядку позиции в никнейме. Позиция в никнейме — порядковый номер символа в конкретном никнейме, нумерация начинается с 1.

Например, если в никнеймах коллег на третьем месте встречаются буквы a и b , то Егор не может использовать буквы a и b в качестве третьего символа своего никнейма, и это правило должно выполняться не только для третьего символа, но и для всех остальных позиций никнейма.

Какой никнейм выберет Егор, если известно, что из всех никнеймов, удовлетворяющих условиям выше, он возьмет лексикографически минимальный?

Входные данные

В первой строке входных данных записаны два целых положительных числа n, k ($1 \leq k \cdot n \leq 10^5$) — количество коллег и длина никнейма. В следующих n строках записаны никнеймы коллег.

Выходные данные

Необходимо вывести строку длины k — никнейм, который выберет Егор. Если невозможно выбрать никнейм, удовлетворяющий всем условиям, выведите «NO» (без кавычек).

Блок «Данные»

Задача 1

Значимые события

Ограничение времени 1 секунда

Ограничение памяти 64Mb

Ввод `devices.json`

Вывод `events.csv`

Каждый сеанс работы БАКа (Большого Адронного Коллайдера) приносит огромное количество данных. Напишите программу, которая поможет их проанализировать и выбрать только значимые события.

Данные с разных датчиков записаны в файле **devices.json**, доступном вашему решению, в формате списка словарей с ключами:

title – обозначение датчика;

events – список данных о событиях, которые зарегистрировал датчик, каждое событие представлено в виде списка из трех значений – *energy*, *momentum*, *direction*.

Выберите события, энергия которых оказалась **выше 15**, и запишите данные о них в файл **events.csv** с заголовками (разделители – запятые):

no, *device*, *energy*, *momentum*, *direction*

номер по порядку, *датчик*, *энергия*, *импульс*, *направление*

События должны быть отсортированы сначала по уменьшению энергии, затем по датчику по алфавиту.

Задача 2

Внутренний туризм-1

База данных содержит информацию о туристических компаниях, базах отдыха и условиях для отдыха в них.

Структура таблиц базы данных:

Recreations (Базы отдыха)

id, *name*, *location_id*, *reservoir*

id, название, *id_расположения*, наличие водоема (0, 1)

Travel_offers (Туристические предложения)

id, *rec_id*, *price*, *duration*, *month*

id, *id_БО*, общая стоимость, продолжительность, месяц начала

Locations (Расположение)

id, *location*

id, местность

Промышленное программирование 9–11 классы ДЕМО

На рисунке показана структура БД и пример заполнения.

The screenshot shows a SQLite database named 'tourism' with three tables: Locations, Receptions, and Travel_offers. The data is as follows:

id	location
1	Sakhalin
2	Altai
3	Baikal
4	Siberia
5	Karelia

id	name	location_id	reservoir
1	Cuffs	2	1
2	Mill	3	1
3	Chemal	4	0
4	Albatros	1	1
5	Hamar	3	1
6	Sheregesh	4	1
7	Viking	2	1
8	Santa	5	1
9	Bear Creek	1	0
10	Barrel	4	1
11	Kaya	2	1
12	Edelweiss	3	0

id	rec_id	price	duration	month
1	1	5	70000	7 August
2	2	2	49000	6 July
3	3	12	140000	14 August
4	4	10	70500	7 August
5	5	1	50000	7 August
6	6	3	100000	12 September
7	7	4	60000	6 August
8	8	4	99000	10 August
9	9	6	71000	7 May
10	10	9	68000	7 August
11	11	5	168000	12 June
12	12	7	79000	8 August
13	13	11	142000	14 August
14	14	9	72000	12 August
15	15	11	50000	6 August

Вам доступна БД в трех вариантах: [sqlite](#), [Excel](#) на 3 листах и [архив](#) из 3 файлов csv.

Определите, для какой местности в БД есть наибольшее количество предложений туров, не дороже 80000 за 7 дней и более и с водоемом. В ответ запишите количество предложений для этой местности, отвечающее заданным условиям.

Задача 3

Внутренний туризм-2

Ограничение времени 1 секунда
Ограничение памяти 256.0 Мб
Ввод стандартный ввод или tourism.db
Вывод стандартный вывод или output.txt

База данных содержит информацию о туристических компаниях, базах отдыха и условиях для отдыха в них.

Структура таблиц базы данных:

Receptions (Базы отдыха)

id, name, location_id, reservoir

id, название, id_расположения, наличие водоема (0, 1)

Travel_offers (Туристические предложения)

id, rec_id, price, duration, month

id, id_БО, общая стоимость, продолжительность, месяц начала

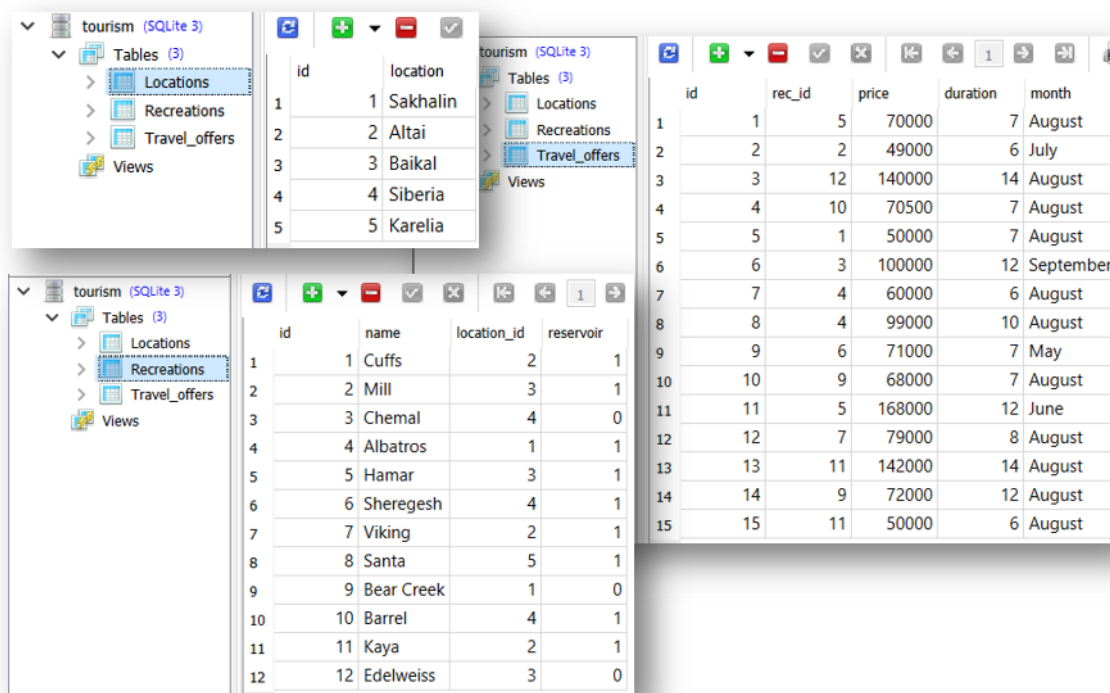
Locations (Расположение)

id, location

id, местность

Промышленное программирование 9–11 классы ДЕМО

На рисунке показана структура БД и пример заполнения.



Напишите запрос, который поможет выбрать все предложения от баз отдыха, для которых выполнены условия:

- есть водоем;
- месяц отдыха – август;
- расположение – Алтай, Байкал или Сибирь

Пример базы данных `tourism.db` для отладки решения можно скачать [здесь](#).

Формат ввода

Ваше решение должно содержать только скрипт, написанный на языке SQL (диалект SQLite), работающий с базой данных `tourism.db`.

Формат вывода

В ответе должно быть 3 столбца, названия не важны, важно, чтобы в первом была база отдыха, во втором стоимость, а в третьем – расположение.

Сортировка в порядке уменьшения цены, затем по базе отдыха по алфавиту.

Примечания

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

name	price	location
Kaya	142000	Altai
Viking	79000	Altai

Промышленное программирование 9–11 классы ДЕМО

name	price	location
Barrel	70500	Siberia
Hamar	70000	Baikal
Cuffs	50000	Altai
Kaya	50000	Altai

Блок «Бэкенд»

Задача 1

Жители Тримунариса

Система Тримунариса очень нестабильна, каждые несколько сотен или тысяч лет происходит катаклизм, приводящий к условиям, несовместимым с жизнью. Обитатели системы на очередном витке эволюции научились записывать информацию о себе и хранить ее на внешней планете.

Информация хранится в текстовом файле **input.txt**, каждая строка которого представляет собой валидный *json*. Каждый словарь *json* содержит обязательные ключи *id* (строка) и *sub_id* (число), а также произвольное количество других ключей и значений по ним.

Напишите сервер и обработку GET-запроса с ключами *id* и *sub_id*, возвращающего словарь в формате JSON с переданными значениями по этим ключам. Если такого словаря нет, нужно вернуть ошибку со статусом 404.

Словарь с ответом возвращается при обращении по адресу (значения по ключам показаны для примера):

http://127.0.0.1:8080/value?id=abc&sub_id=123

Пример входного файла *input.txt*

```
{"id": "abc", "sub_id": 125, "name": "Harry"}
{"id": "abd", "sub_id": 123, "name": "John", "district": "B12"}
{"id": "adc", "sub_id": 123, "age": 185}
{"id": "abc", "sub_id": 123, "mean": "sad"}
```

Ответ сервера

```
{"id": "abc", "mean": "sad", "sub_id": 123}
```

Задача 2

Порядок букв

Тримунарианцы устроены так, что не различают порядок звуков и выражающих их букв. Поэтому для них Том, Мот, Мто, Омт и так далее – обозначают одно и то же слово.

Напишите серверное приложение, которое выберет из текста все анаграммы, сохранит и сможет по запросу вернуть весь список.

Вашему решению будет доступен файл **text.txt** с текстом. Ваш сервер должен по запросу – слову – вернуть список слов из этого текста, которые являются его анаграммами, включая само слово. Порядок алфавитный, регистр не меняется, но не учитывается.

Адрес такого запроса: <http://127.0.0.1:8080/anagram?word='tom'>

Для возвращения словаря со всеми анаграммами (ключ – все буквы слова в нижнем регистре в алфавитном порядке, значение – список слов в форме, в которой они встречаются в тексте, без повторений, порядок алфавитный) используется адрес:

<http://127.0.0.1:8080/anagrams>

Пример файла text.txt

In the deep of the night, a taciturn tale unfolded under the silver glow of the moon. A cat, sleek and stealthy, prowled through the shadows, its eyes glinting like bright stars. Nearby, a dog lay curled up, dreaming of chasing elusive things, while the echoes of the past whispered softly. It was late, and the world was hushed, but the night held secrets only the brave would dare to uncover. As the cat crept closer, it froze, sensing a presence—a god or beast, it could not ascertain. No one knew the history of this place, where time blurred between reality and myth. The dog stirred, its instincts alerting it to something profound in the air, something divine yet terrifying. “On this night,” whispered the wind, “many things shall come to pass.” The cat let out a soft meow, reverberating in the stillness, and the dog barked in response, a simple act of recognition. This moment was a reminder of the delicate balance between companionship and solitude, a fleeting glimpse of the tales that inhabited the corners of their shared existence. They stood together, a testament to loyalty and the mysteries of the night.

Запрос: <http://127.0.0.1:8080/anagram?word=dog>

Ответ: ["dog","god"]

Запрос: <http://127.0.0.1:8080/anagrams>

Ответ: {"act":["act","cat"],"aelt":["late","tale"],"dgo":["dog","god"]}

Задача 3

Погода на третьей планете

На обитаемой третьей планете Тримунариса из-за нестабильной орбиты погода может меняться кардинально по нескольку раз в день. Но с некоторой вероятностью на короткий промежуток времени ее все же можно рассчитать. К сожалению, протуберанец Тримунариса задел спутник, на котором расположены серверы сервиса погоды. И теперь его API иногда может выдавать случайные ошибки, зависает или связь с ним вообще обрывается.

Напишите сервер, который общается с API погодного сервиса по хосту *weather* на порту 9000 и отвечает на запросы на порту 8080.

При обращении к сервису погоды доступны методы:

/temperature?today=...&offset=... – принимает текущую дату в формате '10.05.25' и сдвиг дня, относительно сегодняшнего, на который нужен прогноз (например, для прогноза на сегодня *offset=0*); ответом будет словарь, где ключ – номер часа (строка, с 0), а значение – температура; Пример ответа: {"0": 7, "1": 6, "2": 5, "3": 5, "4": 5, "5": 5, "6": 6}

/precipitation_probability?day=...&start=...&duration=... – принимает дату, час начала прогноза и интервал в часах, для которого необходим прогноз; в ответ сервер сообщит самые неблагоприятные осадки, которые могут быть, и их вероятность в виде строки через двоеточие; Пример ответа: "rain:60"

/cloudless?today=...&offset=...&hour=... – получает текущую дату, сдвиг дня относительно этой даты и час, на который нужен прогноз; возвращает число – вероятность облачности. Пример ответа: "30"

Ваш сервис должен отвечать на запрос **/forecast?day=...** и отдавать прогноз погоды на утро, день, вечер и ночь в формате JSON, где на верхнем уровне есть 4 ключа: *morning, noon, evening u night*.

Утро с 6 часов до 9 (включительно).

День с 10 до 17 (включительно).

Промышленное программирование 9–11 классы ДЕМО

Вечер с 18 до 21 (включительно).

Ночь с 22 до 5 утра следующего дня (включительно).

В каждом интервале надо указать минимальную и максимальную температуру, самые худшие возможные осадки, их среднюю вероятность и значение облачности на начало периода.

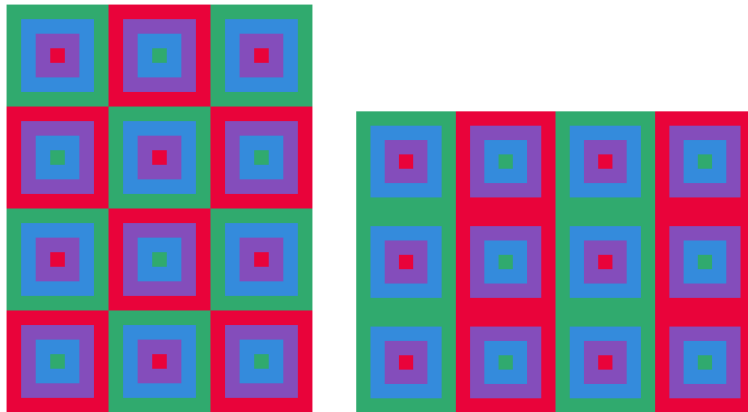
```
{
  "morning": {
    "temperature": {
      "min": 29,
      "max": 35
    },
    "precipitation": {
      "type": "snow",
      "probability": 35
    },
    "cloudiness": 13
  },
  ...
}
```

Блок «Фронтенд»

Задача 1

Плавающие цветные коробки

Сверстайте HTML и CSS для приложенных макетов.



Один файл должен давать соответствующие результаты для любых размеров браузера («резиновая вёрстка»).

Все размеры и цвета нужно взять из файлов макетов — даны для двух размеров окна браузера. Все размеры должны быть заданы в пикселях (px).

Примечания

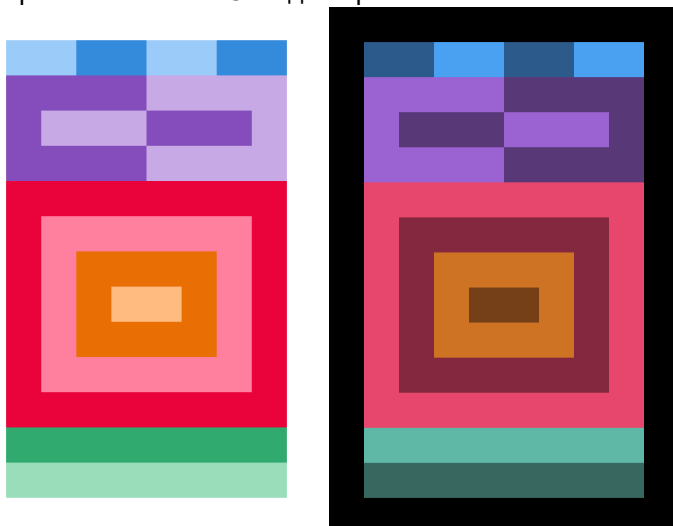
Решение должно представлять из себя один HTML-файл (все нужные стили должны быть включены внутрь).

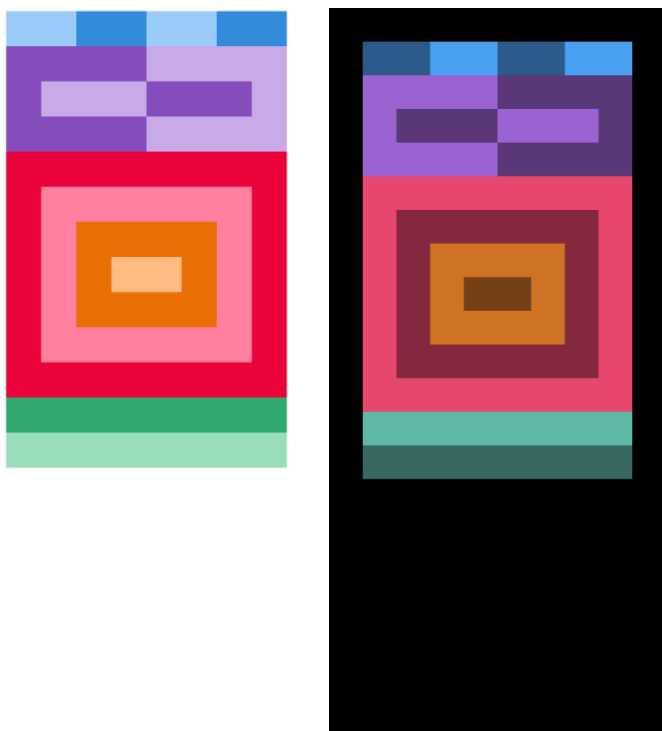
Ваше решение будет тестироваться в последней версии браузера Google Chrome.

Задача 2

Цветные коробки

Сверстайте HTML и CSS для приложенных макетов.





Один файл должен давать соответствующие результаты для любых размеров браузера («резиновая вёрстка»), а также для светлой и тёмной темы оформления (`@media (prefers-color-scheme: dark)`).

Все размеры и цвета нужно взять из файлов макетов — даны для двух размеров окна браузера и для светлой и тёмной темы оформления. Все размеры должны зависеть от размеров окна браузера. Сравнивая разные макеты, можно понять, в каких единицах и каких значений должны быть размеры.

Примечания

Решение должно представлять из себя один HTML-файл (все нужные стили должны быть включены внутрь).

Ваше решение будет тестироваться в последней версии браузера Google Chrome.

Задача 3

Обратный отсчёт

Ограничение времени	1 секунда
Ограничение памяти	64.0 Мб
Ввод	input.json
Вывод	output.txt

Напишите функцию **createCountdown**, которая будет принимать не отрицательное целое число в качестве единственного аргумента и возвращать новую функцию без аргументов. Возвращаемая функция должна, с каждым своим вызовом, по одному, возвращать целые числа от изначально заданного до нуля. Все последующие вызовы, после вызова вернувшего ноль, должны так же возвращать ноль.

Например:

Промышленное программирование 9–11 классы ДЕМО

```
const countdownFrom2 = createCountdown(2)
```

```
countdownFrom2() // 2
```

```
countdownFrom2() // 1
```

```
countdownFrom2() // 0
```

```
countdownFrom2() // 0
```

Если единственный аргумент функции `createCountdown` не является не отрицательным целым числом, это нужно воспринимать как вызов с аргументом равным 0.

Примечания

Решение должно представлять из себя валидный JavaScript с определением функции `createCountdown` на верхнем уровне.