**Требования**

1. Набор используемых технологий ограничен: php, mysql, html, javascript
2. Можно использовать любые javascript библиотеки.
3. Создать публичный репозиторий на github и загрузить туда весь исходный код заданий. В качестве результата передать ссылку на этот репозиторий.

**Задание 1**

Дан массив

[

[399, 9160, 144, 3230, 407, 8875, 1597, 9835],

[2093, 3279, 21, 9038, 918, 9238, 2592, 7467],

[3531, 1597, 3225, 153, 9970, 2937, 8, 807],

[7010, 662, 6005, 4181, 3, 4606, 5, 3980],

[6367, 2098, 89, 13, 337, 9196, 9950, 5424],

[7204, 9393, 7149, 8, 89, 6765, 8579, 55],

[1597, 4360, 8625, 34, 4409, 8034, 2584, 2],

[920, 3172, 2400, 2326, 3413, 4756, 6453, 8],

[4914, 21, 4923, 4012, 7960, 2254, 4448, 1]

]

Среди его ячеек некоторые числа являются числами Фибоначчи *(числами, участвующими в последовательности Фибоначчи: 1, 1, 2, 3, 5, 8, 13, 21).* Найдите сумму чисел Фибоначчи в этом массиве.

**Задание 2**

Возьмите все числа от 1 до 10000 (включительно). Выбросьте из этой последовательности все числа, где встречаются последовательности из трех последовательно восходящих цифр (например 012 или 678). Найдите сумму оставшихся чисел.

**Задание 3**

Создайте таблицу tests и класс для работы с таблицей.

Данные таблицы

id - целое, автоинкрементарное

script\_name - строковое, длиной 25 символов

start\_time - целое

end\_time - целое

result - один вариант из 'normal', 'illegal', 'failed', 'success'

Реализуйте в классе методы:

*fill()*

* доступен только для методов класса
* заполняет таблицу случайными данными

*get()*

* доступен извне класса
* выбирает из таблицы tests, данные по критерию: result среди значений 'normal' и 'success'

**Задание 4**

Предположим, у нас есть код, позволяющий вывести список групп товаров, а также товары, находящиеся в каждой группе. Задача заключается в том, что необходимо добавить к имеющемуся коду функциональность вложенных групп товаров.

В приложенном файле test\_base.sql находится дамп базы данных MySQL, содержащий две таблицы:

1. groups – таблица групп товаров, содержит поля:
   * id – идентификатор группы
   * id\_parent – идентификатор «родительской» группы
   * name – название группы
2. products – таблица товаров, содержит поля:
   * id – идентификатор товара
   * id\_group – идентификатор группы товаров
   * name – название товара

Необходимо написать PHP-скрипт, удовлетворяющий следующим требованиям:

При запуске выводит список (ul) групп товаров первого уровня (groups.id\_parent=0), название каждой группы является ссылкой, которая вызывает этот же скрипт с GET-параметром “group”, равным id группы. Возле названия должно быть написано общее количество товаров в группе (количество товаров в самой группе и во всех ее подгруппах). Пока не выбрана ни одна группа – также выводится список всех товаров.

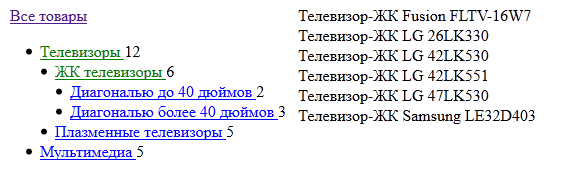
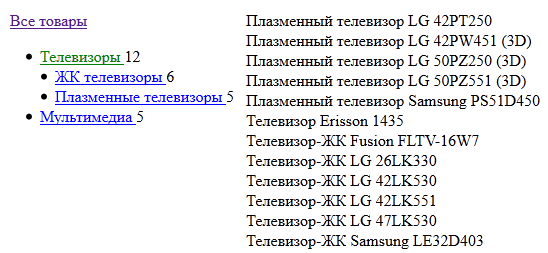
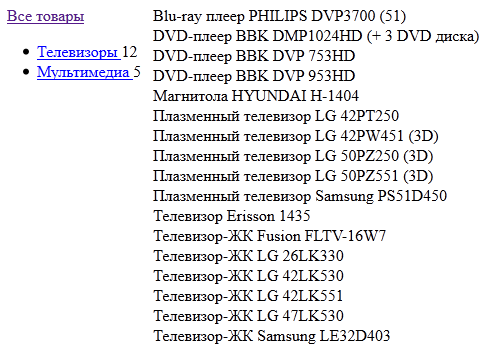
При переходе по ссылке (выборе группы товаров первого уровня) – кроме списка групп товаров первого уровня выводится также список всех подгрупп выбранной группы, все имена подгрупп также являются ссылками, и возле названия подгруппы находится количество товаров, содержащихся в этой подгруппе. Также выводятся все товары, отнесенные к выбранной группе (и всем ее подгруппам).

Количество уровней вложенности групп не ограничено.

Допустимо добавление новых полей в таблицы. Изменять логику работы или типы полей, удалять поля в любой таблице запрещено в целях сохранения обратной совместимости с имеющимся кодом (предполагаем что он есть). В случае добавления полей для всех вновь добавленных полей необходимо подготовить запрос/скрипт, заполняющий их данными, либо описать словами логику работы такого запроса/скрипта.

Никакое специальное оформление не требуется, оцениваться будет логика работы с вложенными группами товаров – sql-запросы и код php, реализующий эту логику, а также html код, реализующий вложенные списки.

«Эталонный» результат работы скрипта:



**Задание 5**

1. Создать страницу с формой. В форме должны быть следующие поля:
   * Имя
   * Фамилия
   * Email
   * Пароль
   * повтор пароля
2. Реализовать отправку этой формы при помощи AJAX.
3. Реализовать обработку AJAX запроса на php. В обработчике нужно:
   * провести валидацию (email содержит @ и пароли совпадают). При желании эти валидации можно также продублировать еще на клиенте (js).
   * задать некий массив уже существующих юзеров (получать его из какой-либо базы данных не требуется). В массиве должны присутствовать поля email, id, name.
   * провести проверку есть ли в этом массиве элемент с заполненным юзером емейлом.
   * результат проверки должен логироваться в файл в любом формате
   * при успешной проверке - форма должна скрываться, а пользователю должно выводиться сообщение об успешной регистрации. При неудачной проверке - пользователю должна выводиться ошибка над формой.
4. Файлы-логи не должны попадать в репозиторий.