Упражнение "Сравнение чисел"

Одна из основных операций с числами – их сравнение. Мы подозреваем, что вы в совершенстве владеете этой операцией и можете сравнивать любые числа.

Необходимо сравнить два целых числа, которые введет пользователь. После этого, нужно вывести на экран сообщение:

• О том, что первое число больше второго.

• О том, что первое число меньше первого.

• О том, что числа равны

Упражнение "Сообщение в рамочке"

Написать программу, которая запрашивает у пользователя строку и выводит ее в консоль в прямоугольнике из звездочек.

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\* Hello, world \*

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Упражнение "Количество жизни"

Необходимо написать программу, которая отобразит пользователю по количеству полных лет, сколько месяцев, дней, минут и секунд он прожил.

Указания:

• Можно предположить, что в году 365 дней.

• Дополнительное задание: приблизительно учесть високосные годы, учитывая, что он бывает 1 раз за 4 года.

Упражнение "Среднее арифметическое"

Создать программу, которая бесконечно спрашивает у пользователя целое число от 1 до 100. Если пользователь вводит не число, пустую строку или не попадающее в критерий число, нужно спросить пользователя заново. Если пользователь нажимает "Отмена", необходимо вывести на экран количество введенных чисел и их среднее арифметическое.

Упражнение "Тест на таблицу умножения".

Здесь мы будем улучшать программу, которая была задана как домашнее задание. Только теперь необходимо генерировать примеры в бесконечном цикле, пока пользователь не нажмет "Отмена". Когда он это сделает, нужно завершить цикл и показать пользователю его статистику - общее количество заданных вопросов, количество правильных и неправильных ответов.

Возможные улучшения и усложнения (необязательно):

• Выбор "сложности" перед началом теста.

• "низкая" сложность будет задавать таблицу умножения на 1 и 2

• "средняя" сложность будет задавать таблицу умножения от 3 до 5

• "высокая" сложность будет задавать всю таблицу, кроме 1, 2 и 5.

• "все" будет задавать всю таблицу умножения.

• Отображение рекомендаций по повторению таблицы в конце теста.

• Например, если пользователь ошибся в задании 9 \* 8. В этом случае, мы будем считать, что он ошибся в таблице на 9 и на 8. Если пользователь в следующий раз ошибется в 8 \* 7, то мы запишем единицу в таблицу на 8 и на 7. Получается, что в таблице на 8 у нас уже будет две ошибки. Необходимо вывести, в какой таблице сколько ошибок было. Если на этом этапе пользователь завершит тест, он увидит:

• В таблице на 9 - 1 ошибка

• В таблице на 8 - 2 ошибки

• В таблице на 7 - 1 ошибка

• Опционально выводить только 3 числа, в которых было допущено максимальное количество ошибок.

Упражнение "Разделение Нлогонии"

После нескольких лет горячих споров о территории, жители страны Нлогонии решили разделить страну на 4 части. Все согласились, что будет определена Точка Раздела, от которой на карте будут проведены две линии - одна с севера на юг, другая с востока на запад. Начиная с самой западной и самой северной, по часовой стрелке, страны будут называться Северозападная Нлогония, Северовосточная Нлогония, Юговосточная нлогония и Югозападная Нлогония.

ООН решило, что будет существовать страница в интернете, где жители смогут определить, в какой стране теперь находятся их дома. Вас наняли реализовать этот проект.

В программе есть один объект (хэш), определяющий точку раздела:

const border = {

x : 12,

y : 6

}

Пользователь должен ввести координаты своего дома с помощью функции prompt.

Вывод: Вы должны вывести результат:

border - если дом находится на границе

NW - если дом находится в северо-западной части страны (NW = North West)

NE - если дом находится в северо-восточной части страны (NE = North East)

SW - если дом находится в юго-западной части страны (SW = South West)

SE - если дом находится в юго-восточной части страны (SE = South East)

Задание "Проверка распределения генератора псевдослучайных чисел".

Компьютер не умеет генерировать по-настоящему случайные числа, поскольку в компьютере все определено очень четко. Однако существуют специальные алгоритмы, которые позволяют генерировать числа, похожие на случайные. По логике вещей генератор случайных чисел должен генерировать числа в соответствии с равномерным распределением. То есть, например, если бы мы подбрасывали монету, то выпадение орла или решки было бы примерно одинаковое количество раз, при достижении статистически значимого результата. Проверим, как работает генератор псевдослучайных чисел Math.random() в JavaScript. Нужно написать программу, которая сгенерирует много случайных чисел от 0 до 99. При этом мы будем разделять их на 10 групп: 0-9, 10-19, 20-29 и так далее. Необходимо вывести, какое количество чисел будет принадлежать каждой группе.

Пример вывода: