# Тестовые задания

## Предисловие

Задания не очень маленькие и не очень простые, мы это понимаем. Но это издержки специфики удаленного сотрудничества: оно требует более тщательной проверки навыков и соображалки разработчика, нежели офисная работа. Поэтому надеемся, что вы нас поймете :) Со своей стороны даем гарантию, что это действительно тестовое задание, а не реальная работа, которую мы хотим получить нахаляву. Мы отлично понимаем, что любой труд должен быть оплачен, поэтому не пользуемся подобными гнусными приемам. Данные задачи получают все разработчики – нам интересно сравнить решения между собой.

В случае, если мы начнем сотрудничество, вы сами будете говорить, сколько хотите за определенную работу и какой это займет срок.

В заданиях намеренно есть некоторая недосказанность, например не указаны точные требования. Это сделано для того, чтобы посмотреть, какое решение вы примите самостоятельно.

Некоторые задания могут показаться довольно «странными» - это нормально :)

Если вы беретесь за выполнение, сообщите об этом, пожалуйста.

Возможно, у вас возникнет вопрос: насколько качественно подходить к решению? Ответ: желательно, чтобы на выходе было решение, которое вам лично очень бы нравилось. А между скоростью разработки и качеством я предпочитаю видеть последнее. Хороший код лучше кратких сроков.

## Задание 1.

Самое объемное.

Напишите программу, которая будет имитировать процедуру automerge в системах контроля исходников.

В чем суть: имеется исходный файл (допустим, программный код), над которым затем поработали 2 программиста, каждый сам по себе. Получилось 2 варианта измененного исходника. Нужно выполнить слияние изменений. Те изменения, которые можно разрулить автоматически, нужно разрулить. Те изменения, которые конфликтуют между собой, нужно проигнорировать, просто поставить пометку «Конфликт» (в реальных системах разработчик конфликтующие изменения разруливает вручную в редакторе, но нам это не нужно).

Пример изменения, которое можно разрулить автоматически: один разработчик добавил функцию, второй в этом месте ничего не изменил.

Пример изменения, которое автоматом нельзя разрулить: оба разработчика изменили одну и ту же строку, но изменили ее по-разному. Соответственно, нельзя однозначно объединить эти два изменения – требуется ручное вмешательство.

Список примеров неполный – остальное предлагается домыслить вам. При этом учитывайте следующие начальные условия:

* Если два изменения разруливаются автоматом, не нужно заморачиваться, будет ли это реально корректно работать. Например, если внутри цикла один разработчик в одном месте добавил строку, другой – добавил свою строку в другом месте, то такое изменение можно разрулить однозначно (добавить обе строки). То, что при этом программа может поломаться, мы в задании не учитываем. Другими словами, никакой проверки синтаксиса не нужно.
* Если один программист удалил строку, а второй с ней ничего не сделал, то мы принимаем это удаление как корректное (т.е. выполняем). Т.е. мы доверяем этому изменению.

Строки считаются равными, если в них совпадают все символы, кроме отступов в начале и конце строки.

Писать на C#, версия фреймворка непринципиальна. Делать как консольное приложение или оконное – неважно, выбираете сами.

На входе: исходный файл; его измененный вариант 1-м программистом; его измененный вариант 2-м программистом.

На выходе: файл с объединенными изменениями.

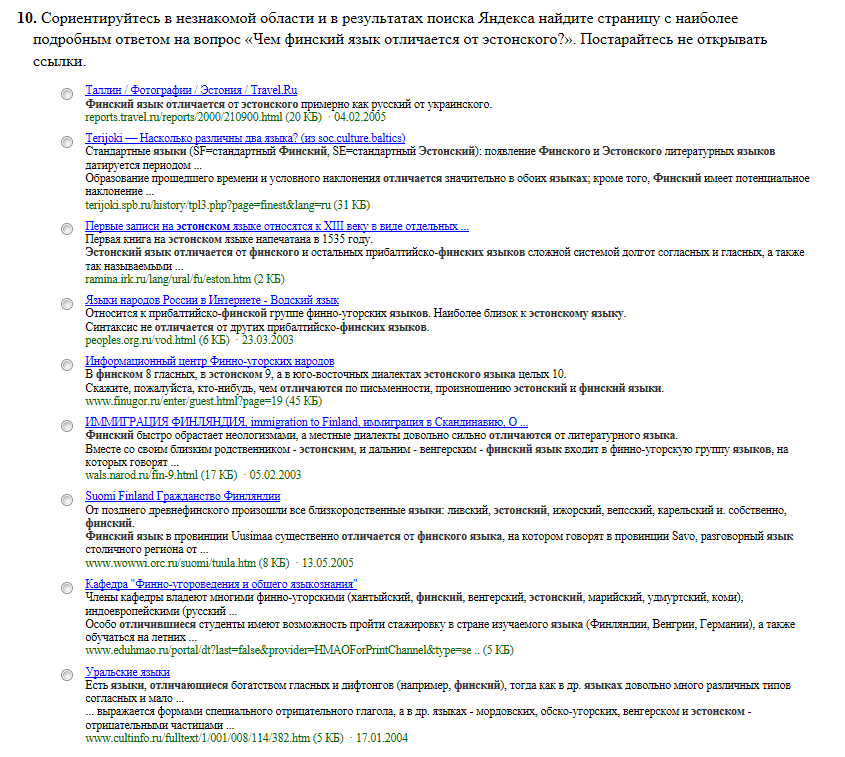
Если беретесь, то оцените предварительно срок решения и сообщите оценку сразу (вместе с подтверждением).

Постарайтесь не пользоваться чужими наработками. Мы знаем, что чаще всего нет смысла изобретать велосипед, но нам интересно посмотреть на ваш код, а не на чужой.

Просьба решение присылать в виде двух архивов: с исполняемыми файлами и с исходниками.

## Задание 2.

Подсмотрено, но интересно:



## Задание 3

Ваша вторая половинка услышала от вас слово «компилятор» и спросила – а что это такое? Как бы кратко вы объяснили?

P.S. Я могу отвечать на присланное решение достаточно долго (до недели) – это нормально. Кол-во присылаемых решений достаточно большое и мне удобней накопить несколько и проверить все сразу. Я отвечаю в любом случае, т.е. молчание не означает отказа – это значит, что я просто еще не смотрел ваше решение.