

Задача 8. Чтение JSON +

Источник:	космической сложности II
Имя входного файла:	<code>input.txt</code>
Имя выходного файла:	<code>output.bin</code>
Ограничение по времени:	разумное
Ограничение по памяти:	разумное

В этой задаче нужно прочитать JSON и записать его в формате BSON. Решение должно работать так же, как в задаче «Чтение JSON», но без использования библиотеки `json-c`. Вы можете использовать решение из той задачи как проверочное при тестировании.

В тестах будут встречаться всевозможные странности, разрешённые в JSON:

1. В строках могут встречаться контрольные символы, записанные в виде `\t`, `\n`, и т.п.
2. В строках могут встречаться символы двойных кавычек и обратного слеша: `\"` и `\"`
3. В строках могут встречаться unicode-символы, записанные в виде `\u5F0a`
4. В строках могут встречаться unicode-символы, записанные в кодировке UTF-8

Эти случаи разрешены спецификацией (лучше её почитать), и `json-c` с ними справляется.

В тексте никогда и ни в каком виде не будет встречаться нулевой байт. Хотя нулевые байты и разрешены спецификацией, в данной задаче они запрещены (и в формате BSON с ними проблема).

Кроме того, в этой задаче могут встречаться вещественные числа, в том числе записанные в экспоненциальной форме, например: `1.345e-7`

Ещё в тестах иногда ставится запятая после последнего элемента в массиве или после последнего поля в объекте. Спецификация запрещает такую вольность, однако многие реализации работают с такими данными.

Наконец, в тестах будут встречаться строчные и блочные комментарии с тем же синтаксисом, как в языке C. Спецификация JSON этого не разрешает, однако `json-c` с этим отлично справляется.

Тесты проверяют описанные выше случаи примерно в порядке перечисления в тексте. Если вы проверили все эти случаи, но решение всё равно не проходит тесты, пишите. Будем смотреть =)