Задача 7. Чтение JSON 2

Источник:	повышенной сложности II
Имя входного файла:	`input.txt`
Имя выходного файла:	`output.txt`
Ограничение по времени:	разумное
Ограничение по памяти:	разумное

В данной задаче нужно самостоятельно реализовать чтение JSON-формата. Чтобы упростить задачу, все тесты соответствуют следующим ограничениям:

- JSON корректен, полностью соответствует спецификации.
- В качестве чисел (number) встречаются только целые числа (без точки, экспоненциального вида и прочей гадости). Отрицательные числа могут быть.
- Внутри каждого строкового значения содержатся только латинские буквы, цифры, знаки препинания и пробелы. Отсутствуют: обратный слэш и контрольные последовательности, всякого рода кавычки, контрольные символы (т.е. ASCII < 32), перевод строки, Unicode-символы вне ASCII.

Чтобы проверить, что вы верно разобрали JSON, вашей программе нужно также обработать запросы. Каждый запрос задаёт путь от корневого объекта или массива JSON-а до искомого значения.

Напоминаем, что структурно JSON состоит из массивов и объектов. Начинать поиск по пути нужно из корня JSON-а. Когда в пути написана строка в двойных кавычках, нужно найти в текущем объекте поле с этим именем и перейти в него. Когда в пути написан индекс в квадратных скобках, нужно найти в текущем массиве элемент с этим индексом и перейти к нему. После выполнения всех переходов поиск останавливается на том значении, которое надо вывести.

Формат входных данных

В первых N строках задаётся сам JSON. Далее идёт отдельная строка, которая содержит в точности строку: `<<<>>>`

После этого идут запросы. Каждый запрос начинается с целого числа K — сколько переходов нужно сделать, чтобы добраться от корня до искомого значения. Далее идёт K строк, в каждой из них прописан один переход: либо строка (имя поля), либо номер в квадратных скобках (индекс в массиве). После всех запросов записано одно число `-1`.

Гарантируется, что размер не превышает 1 мегабайт. Запросы будут составлены таким образом, что если вы будете искать в объекте поле с указанным именем перебором всех полей, то это будет работать достаточно быстро.

Формат выходных данных

Для каждого запроса нужно вывести искомое значение в отдельной строке. Если значение является объектом, нужно вывести `<object>`, а если массивом, то нужно вывести `<array>`. В противном случае нужно вывести само значение в точности так, как оно записано в JSON-е.

Пример

input.txt	output.txt

{
  "array": [
  1,
  2,
  3
  ],
  "boolean": true,
  "color": "#82b92c",
  "null": null,
  "number": 123,
  "object": {
  "a": "b",
  "c": "d",
  "e": [5, {"key": -9}]
  },
  "my string": "Hello World"
}
<<<>>>
4
"object"
"e"
[1]
"key"
1
"null"
1
"my string"
1
"color"
2
"array"
[2]
1
"array"
3
"object"
"e"
[1]
-1
	-9
	null
	"Hello World"
	"#82b92c"
	3
	<array>
	<object>

Комментарий

В данной задаче от вас не требуется извлекать числа, отличать числа от булевых значений, null-ов и строк. Достаточно записать соответствующие значения как строки при парсинге, и потом вывести какие-то из этих строк при запросах.