

C++ assignment: json packer

The task is to read data in JSON format and using dictionary encoding write a compressed stream to disk.

The code should compile with gcc on debian linux.

You may use the following:

- boost
- some JSON parser library
- cmake or simple make

The input will contain records in JSON format. Each record is separated by a new line (LF). The input will only contain simple types, complex and nested data structures will not be present (i.e. map and array).

Input example:

```
{ "key1": "value", "key2": 42, "key3": TRUE }  
{ "sadsf": "dsewtew", "dsre": 3221, "sdfds": "dsfewew" }
```

Basically a record is a list of KVPs (key-value pairs).

Once the input is read, the program should encode the input so that the keys are put into a dictionary. The dictionary maps the keys (strings) to an integer. The key strings are then replaced with a number.

The dictionary would be as follows for the first record:

```
"key1": 1, "key2": 2, "key3": 3
```

This record is then represented as:

```
{ 1: "value", 2: 42, 3: TRUE }
```

Types supported by JSON must be handled and preserved (i.e. integer, string, boolean). The output should be dumped to a file in a binary form using TLV encoding instead of the above text-based representation. This binary TLV encoding is more compact, e.g. a boolean value can be encoded in two bytes (or even less).

At the end of the process the dictionary must be also written to a file using the same encoding method for the string-integer pairs.

The input file can be arbitrary large, make sure to use streaming instead of reading everything into memory. You can assume that the key space (number of distinct key strings) is small and regular in-memory structures (i.e. hashtable) can be used for the dictionary.

While completing this task please try to show your skills (including C++11). Clear, readable and documented code is preferred. Avoid writing obfuscated code, speed should be a secondary objective. Feel free to add unit tests.

Note: This simple assignment shows a very basic algorithm which might be useful to you in the project later but in no way can it be used by us as is.