JSON Parser competition

Specifications

- You must write an executable that will take a (in)valid JSON file, parse it into some JSON data structure and output it in human-readable format.

```
--- example.json
{"minimum":["json","file"],"empty":{}}
--- Output
{
    "minimum": [
        "json",
        "file"
    ],
    "empty": {}
}
```

- Must handle errors:
  - detect invalid json
  - don't crash
  - clean up
  - report error and exit with some nonzero status code
- No memory leaks, you must be able to free your JSON data structure.
- Numeric types will be signed 32 bit integer only.
- String types don't have to support escape characters.
- You don't have to support unicode as described in the RFC.
- You have to write your own JSON parser, so no hyper-optimized library.
- Provide a Makefile with your executable (name=`json_parser.out`, don't forget to compile with optimization flags...

Scoring

- -2 points for every (caught) unhandled parsing error.
- -5 points for not being able to free what you create.
- +3 points for every person below you in the benchmark ranking. Probably we'll use https://github.com/sharkdp/hyperfine on some big file, take the mean and rank everyone based on that.