

YAMLite

截止日期：2018 年 6 月 15 日 23:59

1. 概述

YAML 是“YAML Ain't a Markup Language”的递归缩写，是一个可读性高、用来表达数据序列的格式。YAML 使用类似 Python 的缩进文法，支持多种数据类型，在 YAML 1.2 中，包含了 JSON 的语法，JSON 算是 YAML 1.2 的子集。YAML 特别适合用来表达层次结构式的数据结构，例如配置文件、文件大纲等。

很多编程语言对于 YAML 的解析都有支持。在本次作业中，你需要编写一个解析器，但不需要能够完整地支持 YAML 的所有语法，只需要解析 YAML 的一个子集“YAMLite”，并且能对解析的结果进行查找、转换即可。

本次作业会使用 MOSS 进行查重，如果发现有重复率过高，得 0 分。

请先了解完整的 YAML 语法再进行作业的编写。

参考：

- [1] <http://yaml.org/spec/1.2/spec.html>.
- [2] <https://zh.wikipedia.org/wiki/YAML>.
- [3] <http://www.ruanyifeng.com/blog/2016/07/yaml.html>.

2. YAMLite

YAMLite 是 YAML 的一个子集，它的描述如下：

- a) 由 ASCII 字符集组成；
- b) 每一层的缩进值为两个空格；
- c) 以“#”开始的为注释，注释直到行尾；
- d) 所有的值处在一行，不会出现折行；
- e) 值后面可以有多个空白符；
- f) 支持键值对。键由字母开头，后面跟随零个以上字母、数字、下划线组成，不可由下划线结尾；键和值之间由英文冒号后跟一个空格分割；值可为键值对、数组、字符串、布尔值、整数、浮点数；当值为键值对或者数组时，新增一行，缩进增加两个空格。
- g) 支持数组。数组元素可为键值对、数组、字符串、布尔值、整数、浮点数；数组中的每个元素值跟随在“-”和一个空格后面；当值为键值对或者数组时，新增一行，缩进增加两个空格。
- h) 一个 YAMLite 文件最外层只能是键值对或者数组，不可同时存在；
- i) 值的类型包括键值对、数组、字符串、布尔值、整数、浮点数；字符串由双引号标示，且不包含转义字符；整数最大支持有符号 64 位；浮点数为双精度，且支持科学计数法；

可参考 [sample.yml](#) 进行理解。

请注意，缩进文法属于乔姆斯基一级文法（上下文有关文法），并不是上下文无关文法。

3. 要求

3.1 硬性要求(50 分)

- 如果不能单独完成，可以两个人一组（最后只需交一份），不过，最好是单独完成；
查重后重复率过高的，都得 0 分。
- 在截止日期之前提交；
未能在截止日期前完成的，每拖延一天减 10 分，直至该项为 0 分。
- 不得使用任何第三方库；使用 C/C++、java 编写。
- 使用 C/C++编写的同学，请使用 CMake；使用 java 编写的同学，请使用 Maven 或者 Gradle。

3.2 功能性要求(40 分)

YAMLite 解析程序必须为控制台命令程序，运行示例如下：

```
yamlite [option [value]] file
```

option 为可选参数，file 为待解析的文件的路径。

对于 option 的描述如下：

- option 的值为“-parse”或者不存在时，对输入进行解析，如果无错则输出“valid”；否则输出行号、起始位置以及原因。此项 15 分。

```
yamlite -parse *.yaml 或者 yamlite *.yaml
```

Input 1:

key1: "value1"

key2: "value2"

key3: "value3"

Output 1:

valid

Input 2:

key1: "value1"

key2: "value2"

key3: "value3"

error: "error"

Output 2:

line 4, position 13: expected "<">

- option 的值为“-json”时，将输入转换为等价的 JSON（不需要格式化）输出到同名的.json 文件中。此项 15 分。

```
输入: yamlite -json sample.yaml
```

```
输出: sample.json
```

- option 的值为“-find”时，后跟一个搜索路径，如果搜索成功，将搜索到的值输出，如果值为键值对或者数组，以 JSON 格式（不需要格式化）输出；否则输出“null”。此项 10 分。

```
输入: yamllite -find array[5].key *.yaml
```

```
输出: value
```

搜索路径的语法类似于对象属性访问语法:

- i. 如果是键值对, 路径为键;
 - ii. 如果是数组, 路径为: [数组下标];
 - iii. 如果是上面的组合, 下标直接跟在后面, 键用 “.” 分割
- 例如上面的例子, 意思是搜索 array 的第 5 个元素中键 “key” 对应的值, 也就是 “value”。

3.3 结构性要求 (10 分)

- a) 完整地体现一个解析器的结构, 读取源文件->词法分析->语法分析->简单的语义分析->实现语义; (5 分)
- b) 模块化, 代码设计良好, 可重用性高。(5 分)

4. 提交须知

- 1) 有条件的同学请将程序导出 (或编译) 为可执行文件.jar 或者.exe;
- 2) 程序执行无错请立即退出, 不要在代码里写 `cin.get()` 或者 `system("pause")` 之类的来等待输入退出;
- 3) 编写一个描述你的思路和实现的说明文档, 必须有运行结果截图; 如果是两个人组队, 那么两个人都要写, 并且描述一下自己做了哪些工作;
- 4) 源码放在 src 文件夹下, 可执行文件、文档、测试用例与 src 同级;
- 5) 将所有的文件、文件夹放在.zip 压缩文件中, 文件名格式: 学号-班级-姓名.zip;
如果是组队的, 那么格式为: 成员 1 信息&成员 2 信息.zip
- 6) 提交邮箱: compiler2017@163.com, 收到作业会有自动回复。