

Config File Formats — Practice

CSCI 40

Fall 2026

Total Score: _____/4

How current LLMs scored on this practice bank

2026-08-26

Model	Reasoning	Score
claude-haiku-4.5	low effort	4/6 (67%)
claude-sonnet-5	low effort	4/6 (67%)
claude-fable-5	low effort	5/6 (83%)
claude-opus-4.8	low effort	4/6 (67%)
gpt-5.6-luna	low reasoning	6/6 (100%)
gpt-5.4	low reasoning	6/6 (100%)
gpt-5.6-terra	low reasoning	5/6 (83%)
gpt-5.5	low reasoning	6/6 (100%)
gpt-5.6-sol	low reasoning	6/6 (100%)
gemini-3.1-flash-lite	minimal thinking	1/6 (17%)
gemini-3.5-flash-lite	minimal thinking	3/6 (50%)
gemini-3.6-flash	minimal thinking	2/6 (33%)
gemini-3.7-flash	low thinking	4/6 (67%)

One independent, tool-free prompt per question; deterministic executable ground truth.

JSON-like Configuration Formats

Note. YAML and TOML can represent many of the same dictionaries, lists, strings, numbers, and booleans as JSON. They are not interchangeable in every case because each format supports types or structures that the others cannot represent without changing the data.

1 YAML

Note. YAML stands for *Yaml Ain't a Markup Language*. Python does not have built-in support for YAML; you need to pip install the PyYAML package, and then import the `yaml` module. You can find the full documentation of YAML at <https://yaml.org/>. YAML is famously hated-on by the programmer community; see for example <https://noyaml.com/>.

Problem 1. The following code (circle one)

terminates without error

throws an exception

runs forever

If the code terminates without error, write the output. If the code throws an exception, state the exception.

```
1 import yaml
2 contents=''
3 YAML: YAML Ain't Markup Language
4
5 What It Is:
6     YAML is a human-friendly data serialization
7     language for all programming languages.
8
9 YAML Specifications:
10 - YAML 1.2:
11     - Revision 1.2.2 # Oct 1, 2021 (About this version)
12     - Revision 1.2.1 # Oct 1, 2009
13     - Revision 1.2.0 # Jul 21, 2009
14 - YAML 1.1 # Jan 18, 2005
15 - YAML 1.0 # Jan 29, 2004
16
17 YAML Web Sites:
18     The YAML Company:    yaml.com           # We Support YAML!
19     YAML Playground:     play.yaml.com       # Try many YAMLs!
20     YAML Information:    yaml.info           # Learn about YAML!
21     YAML Test Matrix:    matrix.yaml.info    # Compare YAMLs!
22     Program in YAML:     yamlscrip.org       # Code is Data!
23 ''
24 data = yaml.safe_load(contents)
25
26 a = len(data['YAML Specifications'])
27 print('a=', a)
```

Problem 2. The following code (circle one)

terminates without error

throws an exception

runs forever

If the code terminates without error, write the output. If the code throws an exception, state the exception.

```
1 import yaml
2 contents=''
3 YAML: YAML Ain't Markup Language
4
5 What It Is:
6     YAML is a human-friendly data serialization
7     language for all programming languages.
8
9 YAML Specifications:
10 - YAML 1.2:
11     - Revision 1.2.2 # Oct 1, 2021 (About this version)
12     - Revision 1.2.1 # Oct 1, 2009
13     - Revision 1.2.0 # Jul 21, 2009
14 - YAML 1.1 # Jan 18, 2005
15 - YAML 1.0 # Jan 29, 2004
16
17 YAML Web Sites:
18     The YAML Company:    yaml.com # We Support YAML!
19     YAML Playground:    play.yaml.com # Try many YAMLs!
20     YAML Information:    yaml.info # Learn about YAML!
21     YAML Test Matrix:    matrix.yaml.info # Compare YAMLs!
22     Program in YAML:     yamlscrip.org # Code is Data!
23 ''
24 data = yaml.safe_load(contents)
25
26 b = len(data['YAML Specifications']['YAML 1.2'])
27 print('b=', b)
28
29 c = len(data['YAML Specifications'][0])
30 print('c=', c)
31
32 d = len(data['YAML Specifications'][0]['YAML 1.2'])
33 print('d=', d)
34
35 e = len(data['YAML Web Sites'][0])
36 print('e=', e)
37
38 f = len(data['YAML Web Sites']['The YAML Company'])
39 print('f=', f)
40
41 g = data['YAML on GitHub']['YAML Specs'][0]
42 print('g=', g)
43
44 h = data['What It Is'].split('\n')
45 print('h=', h)
```

Problem 3. The following code (circle one)

terminates without error

throws an exception

runs forever

If the code terminates without error, write the output. If the code throws an exception, state the exception.

```
1 import yaml
2 contents=''
3 name: command_line
4 on:
5   push:
6     branches: ['*']
7 jobs:
8   tests:
9     strategy:
10      matrix:
11        python: [3.11]
12      runs-on: ubuntu-latest
13    steps:
14      - uses: actions/checkout@v2
15      - name: Set up Python ${matrix.python}
16        uses: actions/setup-python@v2
17        with:
18          python-version: ${matrix.python}
19      - name: Run tests
20        run: |
21          pip3 install .
22          markdown-compiler --input_file=README.md
23          markdown-compiler --input_file=README.md --add_css
24    ''
25 data = yaml.safe_load(contents)
26
27 a = len(data['on']['push']['branches'])
28 print('a=', a)
29
30 b = len(data['jobs']['tests']['steps']['name'])
31 print('b=', b)
32
33 c = len(data['jobs']['tests']['steps'][-1]['run'].split('\n'))
34 print('c=', c)
35
36 d = len(data['jobs']['tests']['strategy']['matrix']['python'])
37 print('d=', d)
```

Problem 4. The following code (circle one)

terminates without error

throws an exception

runs forever

If the code terminates without error, write the output. If the code throws an exception, state the exception.

```
1  import yaml
2  contents=''
3  name: command_line
4  on:
5      push:
6          branches:
7              - main
8              - master
9              - testing: true
10 jobs:
11     tests:
12         strategy:
13             matrix:
14                 python:
15                     - 3.9
16                     - 3.10
17                     - 3.11
18     runs-on: ubuntu-latest
19     steps:
20         - uses: actions/checkout@v2
21         - name: Set up Python ${matrix.python}}
22           uses: actions/setup-python@v2
23           with:
24             python-version: ${matrix.python}}
25         - name: Run tests
26           run:
27             pip3 install .
28             markdown-compiler --input_file=README.md
29             markdown-compiler --input_file=README.md --add_css
30     '''
31 data = yaml.safe_load(contents)
32
33 a = len(data['on']['push']['branches'])
34 print('a=', a)
35
36 b = len(data['jobs']['tests']['steps']['name'])
37 print('b=', b)
38
39 c = len(data['jobs']['tests']['steps'][-1]['run'].split('\n'))
40 print('c=', c)
41
42 d = len(data['jobs']['tests']['strategy']['matrix']['python'])
43 print('d=', d)
```

2 TOML

Note. TOML stands for *Tom's Obvious Minimal Language*. Python has built-in support for TOML with the `tomllib` library. You can find the full documentation of TOML at <https://toml.io>.

Problem 5. The following code (circle one)

terminates without error

throws an exception

runs forever

If the code terminates without error, write the output. If the code throws an exception, state the exception.

```
1  import tomllib
2
3  contents = '''
4  title = "TOML Example"
5
6  [owner]
7  name = "Tom Preston-Werner"
8  dob = "1979-05-27T07:32:00-08:00"
9
10 [database]
11 enabled = true
12 ports = [ 8000, 8001, 8002 ]
13 data = [ ["delta", "phi"], [3.14] ]
14 temp_targets = { cpu = 79.5, case = 72.0 }
15
16 [servers]
17
18 [servers.alpha]
19 ip = "10.0.0.1"
20 role = "frontend"
21
22 [servers.beta]
23 ip = "10.0.0.2"
24 role = "backend"
25 '''
26 data = tomllib.loads(contents)
27
28 a = len(data['owner'])
29 print('a=', a)
30
31 b = len(data['owner']['name'])
32 print('b=', b)
33
34 c = len(data['owner'][0])
35 print('c=', c)
36
37 d = len(data['owners']['dob'])
38 print('d=', d)
39
40 e = len(data['database']['data'])
41 print('e=', e)
42
43 f = data['servers']['beta']['ip'][3]
44 print('f=', f)
```

Problem 6. The following code (circle one)

terminates without error

throws an exception

runs forever

If the code terminates without error, write the output. If the code throws an exception, state the exception.

```
1  import toml
2
3  contents = '''
4  # pyproject.toml
5
6  [build-system]
7  requires      = ["setuptools>=61.0.0", "wheel"]
8  build-backend = "setuptools.build_meta"
9
10 [project]
11 name = "realpython-reader"
12 version = "1.0.0"
13 description = "Read the latest Real Python tutorials"
14 readme = "README.md"
15 authors = [{ name = "Real Python", email = "info@realpython.com" }]
16 license = { file = "LICENSE" }
17 classifiers = [
18     "License :: OSI Approved :: MIT License",
19     "Programming Language :: Python",
20     "Programming Language :: Python :: 3",
21 ]
22 keywords = ["feed", "reader", "tutorial"]
23 dependencies = [
24     "feedparser >= 5.2.0",
25     "html2text",
26     'tomli; python_version < "3.11"',
27 ]
28 requires-python = ">=3.9"
29
30 [project.optional-dependencies]
31 dev = ["black", "bumpver", "isort", "pip-tools", "pytest"]
32
33 [project.urls]
34 Homepage = "https://github.com/realpython/reader"
35
36 [project.scripts]
37 realpython = "reader.__main__:main"
38
39 '''
40 data = toml.load(contents)
41
42 a = len(data['project'])
43 print('a=', a)
44
45 b = len(data['project']['dependencies'])
46 print('b=', b)
47
48 c = len(data['project']['urls'])
49 print('c=', c)
50
51 d = len(data['build-system'][-1])
52 print('d=', d)
53
54 e = len(data['build-system']['requires'][-1])
55 print('e=', e)
```