

Languages Isomorphic to JSON

Note 1. In mathematics, the word *isomorphic* means that two objects are “the same in every meaningful way.” (In greek, *iso* means “same” and *morph* means “shape”.) In programming, we call two languages isomorphic if they can be easily converted back and forth between each other without any loss of meaning.

1 YAML

Note 2. 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 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.

```
import yaml
contents=''
YAML: YAML Ain't Markup Language
```

What It Is:

```
YAML is a human-friendly data serialization
language for all programming languages.
```

YAML Specifications:

```
- YAML 1.2:
  - Revision 1.2.2 # Oct 1, 2021 (About this version)
  - Revision 1.2.1 # Oct 1, 2009
  - Revision 1.2.0 # Jul 21, 2009
- YAML 1.1      # Jan 18, 2005
- YAML 1.0      # Jan 29, 2004
```

YAML Web Sites:

```
The YAML Company:  yaml.com           # We Support YAML!
YAML Playground:   play.yaml.com      # Try many YAMLs!
YAML Information:   yaml.info          # Learn about YAML!
YAML Test Matrix:  matrix.yaml.info   # Compare YAMLs!
Program in YAML:   yamlscrip.org      # Code is Data!
'''
```

```
data = yaml.safe_load(contents)
```

```
a = len(data['YAML Specifications'])
print('a=', a)
```

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.

```
import yaml
contents=''
YAML: YAML Ain't Markup Language

What It Is:
  YAML is a human-friendly data serialization
  language for all programming languages.

YAML Specifications:
- YAML 1.2:
  - Revision 1.2.2 # Oct 1, 2021 (About this version)
  - Revision 1.2.1 # Oct 1, 2009
  - Revision 1.2.0 # Jul 21, 2009
- YAML 1.1      # Jan 18, 2005
- YAML 1.0      # Jan 29, 2004

YAML Web Sites:
  The YAML Company:  yaml.com          # We Support YAML!
  YAML Playground:   play.yaml.com     # Try many YAMLs!
  YAML Information:  yaml.info         # Learn about YAML!
  YAML Test Matrix:  matrix.yaml.info  # Compare YAMLs!
  Program in YAML:   yamlscrip.org     # Code is Data!
'''
data = yaml.safe_load(contents)

b = len(data['YAML Specifications']['YAML 1.2'])
print('b=', b)

c = len(data['YAML Specifications'][0])
print('c=', c)

d = len(data['YAML Specifications'][0]['YAML 1.2'])
print('d=', d)

e = len(data['YAML Web Sites'][0])
print('e=', e)

f = len(data['YAML Web Sites']['The YAML Company'])
print('f=', f)

g = data['YAML on GitHub']['YAML Specs'][0]
print('g=', g)

h = data['What It Is'].split('\n')
print('h=', h)
```

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.

```
import yaml
contents=''
name: command_line
on:
  push:
    branches: ['*']
jobs:
  tests:
    strategy:
      matrix:
        python: [3.11]
    runs-on: ubuntu-latest
    steps:
      - uses: actions/checkout@v2
      - name: Set up Python ${matrix.python}
        uses: actions/setup-python@v2
        with:
          python-version: ${matrix.python}
      - name: Run tests
        run: |
          pip3 install .
          markdown-compiler --input_file=README.md
          markdown-compiler --input_file=README.md --add_css
'''
data = yaml.safe_load(contents)

a = len(data['on']['push']['branches'])
print('a=', a)

b = len(data['jobs']['tests']['steps']['name'])
print('b=', b)

c = len(data['jobs']['tests']['steps'][-1]['run'].split('\n'))
print('c=', c)

d = len(data['jobs']['tests']['strategy']['matrix']['python'])
print('d=', d)
```

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.

```
import yaml
contents=''
name: command_line
on:
  push:
    branches:
      - main
      - master
      - testing: true
jobs:
  tests:
    strategy:
      matrix:
        python:
          - 3.9
          - 3.10
          - 3.11
    runs-on: ubuntu-latest
    steps:
      - uses: actions/checkout@v2
      - name: Set up Python ${matrix.python}
        uses: actions/setup-python@v2
        with:
          python-version: ${matrix.python}
      - name: Run tests
        run:
          pip3 install .
          markdown-compiler --input_file=README.md
          markdown-compiler --input_file=README.md --add_css
'''
data = yaml.safe_load(contents)

a = len(data['on']['push']['branches'])
print('a=', a)

b = len(data['jobs']['tests']['steps']['name'])
print('b=', b)

c = len(data['jobs']['tests']['steps'][-1]['run'].split('\n'))
print('c=', c)

d = len(data['jobs']['tests']['strategy']['matrix']['python'])
print('d=', d)
```

2 TOML

Note 7. 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 8. 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.

```
import tomllib

contents = '''
title = "TOML_Example"

[owner]
name = "Tom_Preston-Werner"
dob = "1979-05-27T07:32:00-08:00"

[database]
enabled = true
ports = [ 8000, 8001, 8002 ]
data = [ ["delta", "phi"], [3.14] ]
temp_targets = { cpu = 79.5, case = 72.0 }

[servers]

[servers.alpha]
ip = "10.0.0.1"
role = "frontend"

[servers.beta]
ip = "10.0.0.2"
role = "backend"
'''

data = tomllib.loads(contents)

a = len(data['owner'])
print('a=', a)

b = len(data['owner']['name'])
print('b=', b)

c = len(data['owner'][0])
print('c=', c)

d = len(data['owners']['dob'])
print('d=', d)

e = len(data['database']['data'])
print('e=', e)

f = data['servers']['beta']['ip'][3]
print('f=', f)
```

Problem 9. 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.

```
import toml

contents = '''
# pyproject.toml

[build-system]
requires      = ["setuptools>=61.0.0", "wheel"]
build-backend = "setuptools.build_meta"

[project]
name = "realpython-reader"
version = "1.0.0"
description = "Read the latest Real Python tutorials"
readme = "README.md"
authors = [{ name = "Real Python", email = "info@realpython.com" }]
license = { file = "LICENSE" }
classifiers = [
    "License :: OSI Approved :: MIT License",
    "Programming Language :: Python",
    "Programming Language :: Python :: 3",
]
keywords = ["feed", "reader", "tutorial"]
dependencies = [
    "feedparser>=5.2.0",
    "html2text",
    'tomli; python_version < "3.11"',
]
requires-python = ">=3.9"

[project.optional-dependencies]
dev = ["black", "bumpver", "isort", "pip-tools", "pytest"]

[project.urls]
Homepage = "https://github.com/realpython/reader"

[project.scripts]
realpython = "reader.__main__:main"

'''

data = toml.load(contents)

a = len(data['project'])
print('a=', a)

b = len(data['project']['dependencies'])
print('b=', b)

c = len(data['project']['urls'])
print('c=', c)

d = len(data['build-system'][-1])
print('d=', d)

e = len(data['build-system']['requires'][-1])
print('e=', e)
```