Solutions

Marking: **M1** a correct method, **A1** the correct answer, **B1** an independent mark.

Each answer shows the steps, then the **result**.

2.1

missing `;`: compile; undefined reference: link

segmentation fault: run time; missing header: preprocess

2.2

`main.i`, `main.s`

`main.o`, then the linked program `prog`

2.3

static: each program holds its own copy of the library

so all 40 must be rebuilt and reinstalled

dynamic: replace the one shared library and restart the programs

2.4

(a) a `SyntaxError` when the file loads, before any line runs

(b) no error unless `f()` is called; then a `NameError`

3.1

(a) $200 \times 1.5 + 20 = \mathbf{320}$ s

(b) only that file recompiles: $1.5 + 20 = \mathbf{21.5}$ s

(c) every file that includes it recompiles: $80 \times 1.5 + 20 = \mathbf{140}$ s

(d) $300/8 = 37.5$ s; $+20 = \mathbf{57.5}$ s

3.2

(a) C **0.1** s; CPython **5** s

(b) **50**

(c) e.g. the code is written far faster, and the slow parts can call C libraries such as NumPy

(d) it notices loops that run many times and compiles them to machine code while the program runs

3.3

(a) static 500×12 MB = **6** GB; dynamic 500×0.4 MB = **200** MB

(b) static: rebuild with the fixed library and redeploy 6 GB; dynamic: the servers' normal updates fix the library without touching the program

exercises.lol exercises.com.cn

(c) the program may meet a library version it was not tested with and fail or behave differently ("works on my machine")

(d) e.g. dynamic, with the library version pinned by the servers' package manager, because fixes reach every program at once; or static, if the servers cannot be kept the same, accepting rebuilds for each fix