

Solutions

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

2.1

- (i) **17**
- (ii) **5**
- (iii) [**14**, 17, 21, 19, 18]
- (iv) 21 is at index **2**

2.2

- (i) **6**
- (ii) **9**
- (iii) $6 + 0 + 7 + 4 = \mathbf{17}$

2.3

- first **dequeue** removes **P**
- second **dequeue** removes **Q**
- final queue **R, S**

2.4

- static: e.g. memory is reserved in advance, so it cannot run out while running; direct, fast access by index
- dynamic: e.g. no wasted memory; it can grow to hold any number of items

3.1 B

- the last page visited is the first one returned to: LIFO

3.2 C

- indexes run from 0, so the last row is 9 and the last column 19

3.3 A

- 4 and then 3 are popped from the top

3.4

(a)

- **40**
- the minutes of the second student (row 1) in the third week (column 2)

(b) One answer:

```
for s in range(len(mins)):
    total = 0
    for w in range(len(mins[s])):
        total = total + mins[s][w]
    print(total)
```

- outer loop over the rows
- total reset to 0 for each student
- inner loop adds `mins[s][w]`
- prints once per student, after the inner loop

(c)

- **145, 150, 120**

(d)

- a queue is FIFO: documents print in the order they were sent
- a stack would print the latest first, so an early document could wait for ever