

Solutions

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

2.1

```
for (int v : d)
{
    System.out.println(v);
}
```

- enhanced for header with type `int`
- prints `v` inside the loop

2.2

- indexes 1, 3, 5: **3, 1, 2**

2.3

- when `i` equals `d.length` (6), `d[6]` is **out of bounds**: an `ArrayIndexOutOfBoundsException`

2.4

- the loop variable is a **copy** of each value; assigning to it does not change the array

3.1 B

- visits indexes 3, 2, 1 (stops before 0): $9 + 2 + 7 = 18$

3.2 B

- each element becomes itself plus the **new** previous one: 1, 3, 6, 10

3.3 B

- A starts out of bounds; C misses index 0; D uses values as indexes

3.4

(a)

```
int wetDays = 0;
for (double r : rain)
{
    if (r > 5.0)
    {
        wetDays++;
    }
}
```

- counter set to 0 before the loop
- traverses every element with the test `> 5.0`
- increments the counter inside the `if`

(b)

```
for (int i = 0; i < rain.length; i++)
{
    rain[i] = rain[i] / 10;
}
```

- indexed loop over every element
- assigns `rain[i] / 10` back to `rain[i]`

(c)

- only assigning to `rain[i]` changes the array; an enhanced loop changes only a copy