

Solutions

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

2.1

```
int sum = 0;
for (int i = 1; i <= 99; i += 2) sum += i;
```

- `sum` starts at 0
- loops over the odd numbers 1 to 99 (or tests `i % 2 == 1`)
- adds each to `sum`

2.2

- factors 1, 3, 5, 15: `count` 4
- `largest` 5

2.3

- any all-negative set, such as -4, -9, -2: it gives 0

3.1 B

- multiples of 4: 7; of 6: 5; of both (12, 24): 2; $7 + 5 - 2 = 10$

3.2 B

- one non-positive value sets it `false` for good; C lets a later value reset it

3.3 C

- cast **before** dividing; B divides as `int` first, giving 3.0

3.4

(a)

```
int total = 0;
for (int day = 1; day <= 30; day++) total += getTemp(day);
double avg = (double) total / 30;
```

- loops over days 1 to 30
- adds `getTemp(day)` to a total starting at 0
- divides as a `double`

(b)

```
int hottest = 1;
for (int day = 2; day <= 30; day++)
{
    if (getTemp(day) > getTemp(hottest)) hottest = day;
}
```

- starts with day 1 as the best
- loops over the other days
- compares with the temperature of the best day so far, using `>` so ties keep the earlier day
- stores the **day**, not the temperature