Solutions

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

2.1

- every element is smaller than 25, so `i` reaches 4 = `size()`; 25 is added at **index 4**, the end

2.2

```
ArrayList<String> longWords = new ArrayList<String>();
for (String w : words)
{
    if (w.length() > 5) longWords.add(w);
}
```

- creates a new empty list
- tests `w.length() > 5` for every element
- adds passing elements to the **new** list

2.3

- when `i` equals `size()`, `get(i)` runs **first** and throws an exception; `&&` only skips the second test

2.4

- only the pen is **below** 2.00: **`[pen]`**

3.1 B

- the distinct values are 4, 7, 9

3.2 A

- `i` starts at 2 and counts down from the **original** end, adding 4, 1, 5

3.3 B

- the last index is `size() - 1`, and lists use `get`

3.4

(a)

```
public static String topPlayer(ArrayList<Player> players)
{
    Player best = players.get(0);
    for (Player p : players)
    {
        if (p.getScore() > best.getScore()) best = p;
    }
    return best.getName();
}
```

- starts with the **first** player
- traverses every player
- compares scores with `getScore()` and updates `best`
- returns the **name** of the best player

(b)

```
public static void dropBelow(ArrayList<Player> players, int min)
{
    for (int i = players.size() - 1; i >= 0; i--)
    {
        if (players.get(i).getScore() < min)
        {
            players.remove(i);
        }
        else
        {
            players.get(i).addBonus(10);
        }
    }
}
```

- indexed traversal that does not skip (backwards)
- test `getScore() < min`
- `remove(i)` for low scores
- `addBonus(10)` for the players kept