

Solution: Maintain Total Points for Two Football Teams

This lesson provides a solution to the challenge given in the previous lesson.

WE'LL COVER THE FOLLOWING ^

- Solution
- Solution explanation

Solution

Here is the code that maintains the total points for both teams and finds the winning team.

```
import std.stdio;
int addPoints(int goals1, int goals2,
              ref int points1, ref int points2)
in {
    assert(goals1 >= 0);
    assert(goals2 >= 0);
    assert(points1 >= 0);
    assert(points2 >= 0);

} out (result) {
    assert((result >= 0) && (result <= 2));

} do {
    int winner;

    if (goals1 > goals2) {
        points1 += 3;
        winner = 1;

    } else if (goals1 < goals2) {
        points2 += 3;
        winner = 2;

    } else {
        ++points1;
        ++points2;
        winner = 0;
    }

    return winner;
}
```



```

unittest {
    int points1 = 10;
    int points2 = 7;
    int winner;

    // First team wins
    winner = addPoints(3, 1, points1, points2);
    assert(points1 == 13);
    assert(points2 == 7);
    assert(winner == 1);
    if(winner == 1) {
        writeln(" First team won!");
    }

    // Draw
    winner = addPoints(2, 2, points1, points2);
    assert(points1 == 14);
    assert(points2 == 8);
    assert(winner == 0);
    if(winner == 0) {
        writeln(" It's a draw!");
    }

    // Second team wins
    winner = addPoints(0, 1, points1, points2);
    assert(points1 == 14);
    assert(points2 == 11);
    assert(winner == 2);
    if(winner == 2) {
        writeln(" Second team won!");
    }
}

void main() {
    // ...
}

```



Program to maintain total points for two football teams

Solution explanation

The unittest block is implemented trivially by copying the checks that are already written in `main()`. The only addition above is the test for the case when the second team wins.

Expression-based contracts can be useful for this function:

```

int addPoints(int goals1, int goals2,
              ref int points1, ref int points2)
in (goals1 >= 0)
in (goals2 >= 0)

```

```
in (goals2 >= 0)
in (points1 >= 0)
in (points2 >= 0)
out (result; (result >= 0) && (result <= 2)) {
    // ...
}
```

In the next lesson, you will find a quiz to test your understanding of the concepts covered in this chapter.