

IMG Gaming Coding Exercise

We would like you to provide a coding sample as part of the recruitment process. You should submit a set of files that demonstrate the quality of code you would write as a member of the team. You should complete the test using Scala. The exercise is simple enough that no external libraries should generally be required¹ (though test frameworks are encouraged).

Brief

A new supplier of basketball data is providing scoring data into our system. The data will be streamed to us as matches are being played. We will then parse and clean the data for onward transmission to our clients. The supplier has provided a specification for their format and some sample data. Unfortunately, their format is somewhat obscure and their feed is not always consistent!

Write a utility that:

- can parse incoming data items into a data structure suitable for capturing the state of the match
- maintains a match state as additional data items are received. It should be possible to query the match state (via a method) for the following:
 - the last event (i.e. which team last scored, how many points, at what point through the match and what the resulting match score was)
 - the last n events (where $0 \leq n \leq \text{Total Items}$)
 - all events in the match so far
- handles cases where an invalid data item is received or where data is inconsistent with previous data received.

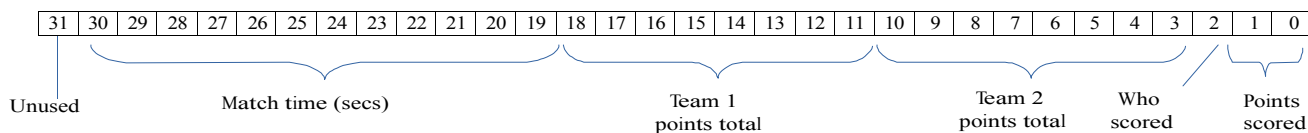
Attached are 2 sample files each containing the data streams for the first 10 minutes of two matches. Your code should contain additional test cases that parse each of the data sets² and validate the events produced. Sample 1's data is self-consistent and should require no special handling. However, sample 2 is not self-consistent. Your utility should interpret the data as best it can such that the set of events is consistent and as accurate as possible. Bear in mind that we need to send data downstream as soon as possible after we have received it.

¹ This is not a strict rule, if you know a library that would help you can use it.

² You do not need to read directly from the files, pasting the data into code is fine

Bit Pattern Specification:

A data event is sent by the supplier each time one of the teams scores. The data contains the match time when the points were scored, an indication of which team scored and the current match score. The data will be encoded into integers using a bit pattern as shown below. The actual integers will be sent using their hexadecimal representations.



Bits	Value	Description
0 - 1	Points scored	Number of points scored (either 1, 2 or 3)
2	Who scored	0 indicates team 1 scored, 1 indicates team 2
3 - 10	Team 2 points total	The total number of points team 2 has scored in the match.
11 - 18	Team 1 points total	The total number of points team 1 has scored in the match.
19 - 30	Elapsed match time	Number of seconds since the start of the match. This represents the match clock, rather than “wall clock” time.

If after 15 seconds of play, Team 1 scores 2 points, then the following will be received:

0x781002 = 7868418 = 0 000000001111 00000010 00000000 0 10

If 15 seconds later, Team 2 replies with 3 points, then the following will be received:

0xf0101f = 15732767 = 0 000000011110 00000010 00000011 1 11

Other sample data points:

0x1310c8a1 - At 10:10, a single point for Team 1 gives them a 5 point lead – 25-20

0x29f981a2 - At 22:23, a 2-point shot for Team 1 leaves them 4 points behind at 48-52

0x48332327 - At 38:30, a 3-point shot levels the game for Team 2 at 100 points each