This description is based on that at www.xprogramming.com/xpmag/acsBowling.htm

Problem Description

Create a program, which, **given a valid sequence of rolls** for one **line** of American Ten-Pin Bowling, produces the total score for the game. Here are some things that the program will not do:

- We will not check for valid **rolls**.
- We will not check for correct number of **rolls** and **frames**.
- We will not provide scores for intermediate **frames**.

Depending on the application, this might or might not be a valid way to define a complete story, but we do it here for purposes of keeping the kata light. I think you'll see that improvements like those above would go in readily if they were needed for real.

We can briefly summarize the scoring for this form of bowling:

- Each game, or "**line**" of bowling, includes ten **turns**, or "**frames**" for the bowler.
- In each frame, the bowler gets up to two tries to knock down all the pins.
- If in two tries, he fails to knock them all down, his score for that **frame** is the total number of pins knocked down in his two tries.
- If in two tries he knocks them all down, this is called a "**spare**" and his score for the **frame** is ten plus the number of pins knocked down on his next throw (in his next turn).
- If on his first try in the **frame** he knocks down all the pins, this is called a "**strike**". His turn is over, and his score for the **frame** is ten plus the simple total of the pins knocked down in his next two **rolls**.
- If he gets a **spare** or **strike** in the last (tenth) **frame**, the bowler gets to throw one or two more **bonus** balls, respectively. These **bonus** throws are taken as part of the same turn. If the **bonus** throws knock down all the pins, the process does not repeat: the **bonus** throws are only used to calculate the score of the final **frame**.
- The game score is the total of all **frame** scores.

More info on the rules at: www.topendsports.com/sport/tenpin/scoring.htm

Clues

What makes this game interesting to score is the lookahead in the scoring for **strike** and **spare**. At the time we throw a **strike** or **spare**, we cannot calculate the frame score: we have to wait one or two **frames** to find out what the **bonus** is.

Suggested Test Cases

(When scoring "X" indicates a **strike**, "/" indicates a **spare**, "-" indicates a **miss**)

- "XXXXXXXXXXXX" (12 rolls: 12 strikes) = 10+10+10 + 10+10+10 + 10+10+10 + 10+10+10 + 10+10+10 + 10+10+10 + 10+10+10 + 10+10+10 + 10+10+10 + 10+10+10 = 300
- "9-9-9-9-9-9-9-9-9-9-" (20 rolls: 10 pairs of 9 and miss) = 9 + 9 + 9 + 9 + 9 + 9 + 9 + 9 + 9 + 9 = 90
- "5/5/5/5/5/5/5/5/5/5/5" (21 rolls: 10 pairs of 5 and spare, with a final 5) = 10+5 + 10+5 + 10+5 + 10+5 + 10+5 + 10+5 + 10+5 + 10+5 + 10+5 + 10+5 = 150

Comments from those who have mastered this Kata

Write some thoughts here about what you have learnt from this Kata. You don't have to post all the code of your solution - I think the solution in itself is less interesting than the path you took to get there and what decisions you made. Just seeing the code won't necessarily help me to reproduce it for myself. So in this section various people might go through the main parts of the problem and how they tackled them, what design ideas were discarded, and which order the test cases were implemented in.

- One interesting point to note is that without counting frames in any way (although I don't think this was intended as a 'hard' requirement for the initial Kata completion), finding an elegant way to identify the end of the game/last "real" frame becomes difficult (ie: assuming there are final 'bonus' rolls included in a given test case).
- **Update**: After trying various things, including writing out a logic matrix for possible end-of-game combinations, I'm not sure it's possible to detect whether a final 'throw' counts as bonus-only or as part of an actual frame, unless you're counting frames.