

---

## Problem #2: Bowling for Dummies

**Program Name:** bowling.java

**Input File:** bowling.dat

Bowling is a fun game for all ages. The object of the game, like so many others, is to get the highest score possible. Points are scored by rolling a ball down a lane and knocking down ten pins that are arranged in a triangular pattern.

A game is scored by summing the scores for each of ten *frames*, where each frame a player is given up to two rolls to knock down the pins. If a player knocks down all the pins on the first roll of a frame, the frame is scored as a strike and marked with an 'X'. A strike is worth ten points plus one point for each pin knocked down in the next two rolls. If a player knocks down all remaining pins in the second roll of a frame, the frame is scored as a spare and marked with a '/'. A spare is worth ten points plus one point for each pin knocked down in the next roll. If the player does not knock down all the pins in the two rolls for a frame, the score for the frame is the total number of pins knocked down for that frame.

The tenth frame is an exception to the above rules and its score is the total number of pins knocked down. Unlike the other frames, if a player scores a strike on his first roll for the tenth frame, he is allowed two more rolls for the frame, or if he scores a spare on his first two rolls of the frame, he is allowed one more roll for the frame.

Given the above scoring method, the highest possible score for a bowling game is 300.

### Input

The first line will be a single integer,  $n$ , indicating the number of games to be scored.

Each of the next  $n$  lines will represent a single game and will consist of all the rolls for that game. The outcome of each roll will be indicated with an integer representing the number of pins knocked down, a '/' for a spare, or a 'X' for a strike.

### Output

Output will consist of  $n$  lines, where each line is the score of a bowling game from the input. Game scores should be listed in the order of the games in the input.

### Example Input File

```
3
8 / 9 / X X 3 6 7 / 8 1 4 / X X 6 /
8 1 9 0 X X 3 6 7 2 8 1 4 5 9 / 3 / 2
X X X X 9 / X 8 1 X 7 / X X X
```

### Example Output To Screen

```
183
121
227
```