
CricCenter

Objective

Write a C program to compute statistics of a cricket T20 inning. The program must parse a string of events, calculate the team score and run rate, log the fall of wickets, track the state of the batters, and generate a scorecard.

For those unfamiliar with the game of cricket, refer to the [appendix](#) for details.

Specifications

Your program should read the match data from two lines of input:

1. Batting Order

- A space-separated ordered list of exactly 11 batter names.
- Each name is a single word and has a maximum length of 9 characters.

2. Event String

- A space-separated sequence of tokens representing the outcome of a ball.
- The list of valid tokens includes: 0, 1, 2, 3, 4, 6, W, d, n

Tokens

- 0 → dot-ball: No runs scored on the ball
- 1, 2, 3 → Runs scored by running between wickets
- 4, 6 → Ball reaches the boundary along the ground or in the air
- W → Striker is dismissed and a wicket falls
- d, n → wide and no-ball: 1 run added and ball re-bowled

Rules

- Before the inning begins, all batter statistics are initialized to 0. The first two batters in the order open the batting as striker and non-striker, respectively.
- Each batter's status is **Yet to Bat** until they enter the field. Once they do, it becomes **Not Out**, and upon dismissal, it changes to **Out**. Batters who are **Not Out** are marked with an asterisk * next to their runs in their individual score.
- Runs are added based on the token. 0 and W result in no runs. 1, 2, 3, 4, 6 are added to both the team total and the striker's total. d and n add 1 run each to the team total only and not to the striker.
- A ball is considered legal if it is neither a d nor a n. Illegal balls are re-bowled, do not count as balls faced by the batter, and do not change the strike.
- The striker and non-striker swap ends when 1 or 3 runs are scored, or at the end of an over. If 1 or 3 runs are scored off the last ball of an over, the striker retains the strike.

- A **W** always means the striker is out, and the next batter in the batting order replaces them. A wicket should be recorded on the ball on which it occurs.
- Overs are represented as **O.B**, where **O** is the number of completed overs ranging from 0 to 20 and **B** is the number of balls in the current over ranging from 0 to 5. When **B** reaches 6, **O** is incremented by 1 and **B** is reset to 0.
- For instance, 8.4 denotes 8 overs and 4 balls. After 2 more legal balls have been bowled, it becomes 9.0 overs.
- The inning ends when 10 wickets have fallen (wickets numbered 1 to 10) or 120 legal balls (20 overs * 6 balls) have been bowled. Note that the actual number of balls may be more than 120 due to illegal balls.

Calculations

- Overs can be tracked by maintaining the number of legal balls. **O** and **B** can be derived using $O = \text{legal_balls} / 6$ and $B = \text{legal_balls} \% 6$ at any point.
- Run rate is calculated as $(\text{total_runs} / \text{legal_balls}) * 6$. The value is formatted to two decimal places.
- Strike rate of a batter is defined as $(\text{runs_scored} / \text{balls_faced}) * 100$. Strike rate is 0 when balls faced is 0. The value is formatted to two decimal places.

Data Structures

Team Statistics (Variables)

Track the team statistics using the following variables. The names may be changed.

```
int total_runs, total_wickets, legal_balls;

float run_rate;
```

Fall of Wickets (Linked List):

Maintain a linked list to track the fall of wickets. Each time a wicket is parsed, capture the current score and overs, allocate a node, and append it to the list.

```
struct FOW
{
    int wicket_number, team_score, overs, balls;
    struct FOW *next;
};

struct FOW *fow_head, *fow_tail;
```

Batters (Array of Structures)

Define a structure to store individual batter statistics. Declare an array of size 11 to represent the ordered team roster with each element corresponding to one batter.

```
struct Batter
{
    char name[10], status[12];
    int runs_scored, balls_faced, fours, sixes;
    float strike_rate;
};

struct Batter batters[11];
```

Strike Rotation (Pointers)

Use two pointers to track the batters on the field. Update them when the strike changes or a wicket falls.

```
struct Batter *striker, *non_striker;
```

Commands

The program should print statistics based on commands. Examples below are for 3 overs, but the test cases will cover a full inning (up to 20 overs or 10 wickets).

1. TEAM_SCORE

[6 Test Cases × 0.5 Marks = 3 Marks]

Print the total runs scored by the team along with wickets lost.

```
<total_runs>-<total_wickets>
```

Sample Input:

```
Rohit Gill Virat Shreyas Rahul Surya Jadeja Ashwin Shami Bumrah Siraj
1 0 4 d 2 3 1 W 1 0 0 n 1 2 1 1 0 6 W 6
TEAM_SCORE
```

Sample Output:

```
31-2
```

Explanation:

A total of 31 runs were scored including extras and 2 wickets were lost.

2. RUN_RATE

[4 Test Cases × 0.5 Marks = 2 Marks]

Print the run rate calculated using total runs and overs.

```
<run_rate>
```

Sample Input:

```
Rohit Gill Virat Shreyas Rahul Surya Jadeja Ashwin Shami Bumrah Siraj  
1 0 4 d 2 3 1 W 1 0 0 n 1 2 1 1 0 6 W 6  
RUN_RATE
```

Sample Output:

```
10.33
```

Explanation:

Run rate is calculated as $(31 / 18) * 6 = 10.33$, formatted to two decimal places.

3. FALL_OF_WICKET <wicket_number>

[10 Test Cases × 0.5 Marks = 5 Marks] (50% penalty if linked lists are not used)

Traverse the FOW list and print the details in the specified format.

```
<wicket_number>-<team_score> (<overs>.<balls>)
```

Sample Input:

```
Rohit Gill Virat Shreyas Rahul Surya Jadeja Ashwin Shami Bumrah Siraj  
1 0 4 d 2 3 1 W 1 0 0 n 1 2 1 1 0 6 W 6  
FALL_OF_WICKET 1  
FALL_OF_WICKET 2
```

Sample Output:

```
1-12 (1.1)  
2-25 (2.5)
```

Explanation:

The 1st and 2nd wickets fell on the 1st ball of the 2nd over (7th legal ball) and the 5th ball of the 3rd over (17th legal ball), at scores of 12 and 25 runs, respectively.

4. BATTER_SCORE <batter_name>

[14 Test Cases × 0.5 Marks = 7 Marks] (50% penalty if arrays are not used)

Traverse the batter array and print the runs scored and balls faced by the batter.

```
<runs>[*] (<balls>)
```

Sample Input:

```
Rohit Gill Virat Shreyas Rahul Surya Jadeja Ashwin Shami Bumrah Siraj
1 0 4 d 2 3 1 W 1 0 0 n 1 2 1 1 0 6 W 6
BATTER_SCORE Rohit
BATTER_SCORE Shreyas
BATTER_SCORE Jadeja
```

Sample Output:

```
2 (3)
6* (1)
0 (0)
```

Explanation:

Rohit scored 2 runs and was dismissed on his third ball. Shreyas hit a six off his first ball and remained not out. Jadeja never came out to bat.

5. SCORECARD

[6 Test Cases × 0.5 Marks = 3 Marks]

Generate and print an HTML table containing the statistics of all 11 batters.

```
<style>.scorecard th,td {border:1px solid;padding:6px}</style>
<table class="scorecard">
<tr><th>Name</th><th>Status</th><th>Runs</th><th>Balls</th><th>Fours</t
h><th>Sixes</th><th>Strike Rate</th></tr>
<tr><td>name</td><td>status</td><td>runs_scored</td><td>balls_faced</td
><td>fours</td><td>sixes</td><td>strike_rate</td></tr>
</table>
```

Sample Input:

```
Rohit Gill Virat Shreyas Rahul Surya Jadeja Ashwin Shami Bumrah Siraj
1 0 4 d 2 3 1 W 1 0 0 n 1 2 1 1 0 6 W 6
SCORECARD
```

Sample Output:

```
<style>.scorecard th,td {border:1px solid;padding:6px}</style>
<table class="scorecard">
<tr><th>Name</th><th>Status</th><th>Runs</th><th>Balls</th><th>Fours</t
h><th>Sixes</th><th>Strike Rate</th></tr>
<tr><td>Rohit</td><td>Out</td><td>2</td><td>3</td><td>0</td><td>0</td><
td>66.67</td></tr>
<tr><td>Gill</td><td>Out</td><td>17</td><td>11</td><td>1</td><td>1</td>
<td>154.55</td></tr>
<tr><td>Virat</td><td>Not
Out</td><td>4</td><td>3</td><td>0</td><td>0</td><td>133.33</td></tr>
<tr><td>Shreyas</td><td>Not
Out</td><td>6</td><td>1</td><td>0</td><td>1</td><td>600.00</td></tr>
<tr><td>Rahul</td><td>Yet to
Bat</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0.00</td></tr>
<tr><td>Surya</td><td>Yet to
Bat</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0.00</td></tr>
<tr><td>Jadeja</td><td>Yet to
Bat</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0.00</td></tr>
...
</table>
```

Name	Status	Runs	Balls	Fours	Sixes	Strike Rate	Surya	Yet to Bat	0	0	0	0	0.00
Rohit	Out	2	3	0	0	66.67	Jadeja	Yet to Bat	0	0	0	0	0.00
Gill	Out	17	11	1	1	154.55	Ashwin	Yet to Bat	0	0	0	0	0.00
Virat	Not Out	4	3	0	0	133.33	Shami	Yet to Bat	0	0	0	0	0.00
Shreyas	Not Out	6	1	0	1	600.00	Bumrah	Yet to Bat	0	0	0	0	0.00
Rahul	Yet to Bat	0	0	0	0	0.00	Siraj	Yet to Bat	0	0	0	0	0.00

Explanation:

Rohit, Gill, Virat and Shreyas scored 2 (3), 17 (11), 4* (3) and 6* (1), respectively. Their strike rates were calculated as (runs/balls) * 100 and formatted to two decimal places: $2/3 * 100 = 66.67$, $17/11 * 100 = 154.55$, $4/3 * 100 = 133.33$ and $6/1 * 100 = 600.00$. The remaining batters never batted and have scores of 0 (0).

Guidelines

- Marks are allocated per command as specified. There are 40 test cases across all commands, each carrying 0.5 marks.
- Each test case contains exactly one command, but may include one or more instances of that command.
- Use the provided starter template to simplify the implementation. This is optional, and grading will be based solely on the final output. However, the use of linked lists and arrays is mandatory for the respective commands.
- Follow a step-by-step approach:
 1. Parse the event string and compute basic team statistics such as total runs, wickets, and legal balls.
 2. Use the computed runs, wickets, and overs to calculate the run rate.
 3. Record the fall of wickets using a linked list by capturing the match state whenever a wicket is encountered.
 4. Implement striker and non-striker pointers to track the batters on the field and update individual batter statistics in the batter array.
 5. Compute strike rates and traverse the batter array to generate the HTML scorecard.
- Build and test each step incrementally before proceeding to the next.
- You can assume that all test cases satisfy the given problem constraints, and you are not required to handle invalid test cases.

Appendix

Cricket is a bat-and-ball game played between two teams of eleven players. The game is played on a field marked by a circular boundary. At the center of the field is a pitch with stumps at each end.

- Two players from the batting team are on the field at any time: the striker, who faces the ball, and the non-striker, who stands at the other end.
- A player from the fielding team, called the bowler, bowls the ball towards the striker. Balls are grouped into overs, each with 6 legal balls.
- The striker attempts to hit the ball and score runs. One run is scored when the two batters swap ends by running. Additional runs can be scored by hitting the ball to the boundary along the ground for four or in the air for six.
- The fielding team tries to prevent runs and dismiss batters by taking wickets. When a batter loses their wicket, they leave the field, and the next batter in the batting order comes in.
- An inning continues until 10 wickets are taken or the stipulated number of overs is completed. In a T20 match, each team can bat for 20 overs.