### Problem Statement: **Cricket Scoreboard**

You are tasked with creating a basic **Cricket Match Scoreboard** system. The system should allow you to:

1. Add players to teams.
2. Record runs and wickets for the match.
3. Display the current score and match summary.

#### Classes

1. **Player** (2 Marks)
   - Represents a player with attributes:
     - `name`: Name of the player.
     - `runs`: Total runs scored by the player.
   - Methods:
     - `__str__`: Returns the player's name and runs in the format: `"Player Name: X runs"`.

2. **Team** (3 Marks)
   - Represents a cricket team.
   - Attributes:
     - `team_name`: Name of the team.
     - `players`: List of `Player` objects in the team.
   - Methods:
     - `add_player`: Adds a new player to the team.
     - `__str__`: Returns a summary of the team's players and their runs.

3. **Match** (5 Marks)
   - Represents a simple cricket match.
   - Attributes:
     - `team`: The team currently batting.
     - `score`: Total runs scored by the team.
     - `wickets`: Number of wickets fallen.

  - Methods:
    - `record_run`: Adds runs to the team's score.
    - `record_wicket`: Increases the wicket count.
    - `summary`: Displays the team's total score, wickets fallen, and each player's performance.

### Example Interaction

#### Sample Scenario
1. A match starts with **Team A** batting.
2. Players are added to the team.
3. Runs and wickets are recorded.
4. Display the match summary at the end.

#### Example Code
```python
# Create a Team
team_a = Team("Team A")

# Add players to the team
team_a.add_player(Player("Player A1"))
team_a.add_player(Player("Player A2"))

# Start a match with Team A batting
match = Match(team_a)

# Record some runs and wickets
match.record_run("Player A1", 4)  # Player A1 hits a boundary
match.record_run("Player A1", 1)  # Player A1 scores a single
match.record_wicket("Player A1")  # Player A1 is out

match.record_run("Player A2", 6)  # Player A2 hits a six

# Display match summary
match.summary()
```

### Mark Distribution

1. **Player Class** (2 Marks): Correct implementation of the `Player` class and its `__str__` method.

2. **Team Class** (3 Marks): Proper handling of players in the team and its `add_player` and `__str__` methods.

3. **Match Class** (5 Marks): Complete functionality for recording runs, wickets, and displaying the match summary.

### Challenge Task (Optional Bonus)

1. Track the number of balls faced by each player.

2. Display the strike rate for each player in the summary.