

題目：

跟期中考題目大致相同，但部分規則有所改變

- 進壘與得分
 - 原則上：如果一壘安打，則壘包上的人個前進一個壘包，如果三壘有人，則會得一分。依此類推，如果二壘安打，壘包上的人會前進兩個壘包。
 - 如果特別標示得分 R_n ，表示得分狀況特殊，例如一壘安打造成 2, 3 壘都回來得分，此時特別標記 1B 2R
- 雙殺與三殺
 - 滾地球可能產生雙殺甚至於三殺，請修改速記來標示，並計算出局數
- 整場速記
 - 在單局速記測試完成後，接著完成上半場與下半場的速記，也就是能顯示客隊與主隊差異。
 - 接著完成九局（或延長）的速記，注意球員的上場應具備延續性，一局上如果是 1-3 棒打擊，二局上應該由第 4 棒開始。

開發與測試時應注意

- 模組化開發，方便測試。
- 對於異常的速記要能夠偵測，例如壘包上只有一人，安打後卻得到三分，顯然不合理。
- 多利用 JUnit 的功能，例如參數化測試等。

程式、執行畫面及其說明：

程式碼擷取規則有改變的部分。

- 進壘與得分
 - 如果一壘安打，則壘包上的人個前進一個壘包，如果三壘有人，則會得一分。依此類推，如果二壘安打，壘包上的人會前進兩個壘包。

```
private static int advanceRunners(boolean[] bases, int baseAdvance, boolean isBBorE) {
    int autoRuns = 0;
    boolean[] next = new boolean[3];

    // 先處理原本壘上的跑者
    for (int i = 2; i >= 0; i--) {
        if (!bases[i]) continue;
        int originBase = i + 1;
        int newBase = originBase + baseAdvance;
```

```

        int newBase = originBase + baseAdvance;
        if (newBase > 3) {
            autoRuns++;
        } else {
            next[newBase - 1] = true;
        }
    }

    // 打者的去處
    int batterDest;
    if (baseAdvance == 4) {
        batterDest = 4; // 全壘打
    } else if (isBBorE) {
        batterDest = 1; // 保送 / 失誤 → 上一壘
    } else {
        batterDest = baseAdvance; // 1B / 2B / 3B
    }

    if (batterDest > 3) {
        autoRuns++;
    } else if (batterDest >= 1) {
        next[batterDest - 1] = true;
    }

    // 更新壘包
    System.arraycopy(next, srcPos: 0, bases, destPos: 0, length: 3);
    return autoRuns;
}

```

- 如果特別標示得分 R_n ，表示得分狀況特殊，例如一壘安打造成 2, 3 壘都回來得分，此時特別標記 1B 2R

```

int explicitRuns = extractRuns(event);

```

```

//  $R_n$ 
int addRuns;
if (explicitRuns > 0) {
    // 有標示  $R_n$ 
    addRuns = explicitRuns;
} else {
    // 沒標示  $R_n$ 
    addRuns = autoRuns;
}
runs += addRuns;

```

- 雙殺與三殺

- 滾地球可能產生雙殺甚至於三殺，請修改速記來標示，並計算出局數

```
private static int outsFromGroundPlay(String event) { 1 usage new *
    String firstToken = event.split(regex: " ")[0];
    String digitsOnly = firstToken.replaceAll(regex: "[^0-9]", replacement: "");
    int len = digitsOnly.length();
    return Math.max(1, len - 1);
}

else if (event.matches(regex: "[0-9].*?-.*")) {
    int addOuts = outsFromGroundPlay(event);
    outs = Math.min(3, outs + addOuts);
    batterIndex++;
}
```

- 整場速記

- 在單局速記測試完成後，接著完成上半場與下半場的速記，也就是能顯示客隊與主隊差異。

```
// ----- 局上：客隊 -----
System.out.println("----- 第 " + inningNo + " 局上：客隊 " + away.name + " 進攻 -----");
if (topEventsByInning != null && i < topEventsByInning.length
    && topEventsByInning[i] != null) {
    HalfInningResult r = simulateHalfInning(
        away.lineup, away.batterIndex, topEventsByInning[i], logExtraAfter3Out: true);
    away.batterIndex = r.nextBatterIndex;
    away.inningRuns[i] = r.summary.runs;
    away.totalRuns += r.summary.runs;
    show(r.summary);
} else {
    System.out.println("(本局無速記事件)");
}

// ----- 局下：主隊 -----
System.out.println("----- 第 " + inningNo + " 局下：主隊 " + home.name + " 進攻 -----");
if (botEventsByInning != null && i < botEventsByInning.length
    && botEventsByInning[i] != null) {
    HalfInningResult r = simulateHalfInning(
        home.lineup, home.batterIndex, botEventsByInning[i], logExtraAfter3Out: true);
    home.batterIndex = r.nextBatterIndex;
    home.inningRuns[i] = r.summary.runs;
    home.totalRuns += r.summary.runs;
    show(r.summary);
} else {
    System.out.println("(本局無速記事件)");
}
```

- 接著完成九局（或延長）的速記，注意球員的上場應具備延續性，一局上如果是 1-3 棒打擊，二局上應該由第 4 棒開始。

```

// ----- 局上：客隊 -----
System.out.println("----- 第 " + inningNo + " 局上：客隊 " + away.name + " 進攻 -----");
if (topEventsByInning != null && i < topEventsByInning.length
    && topEventsByInning[i] != null) {
    HalfInningResult r = simulateHalfInning(
        away.lineup, away.batterIndex, topEventsByInning[i], logExtraAfter3Out: true);
    away.batterIndex = r.nextBatterIndex;
    away.inningRuns[i] = r.summary.runs;
    away.totalRuns += r.summary.runs;
    show(r.summary);
} else {
    System.out.println("(本局無速記事件)");
}

// ----- 局下：主隊 -----
System.out.println("----- 第 " + inningNo + " 局下：主隊 " + home.name + " 進攻 -----");
if (botEventsByInning != null && i < botEventsByInning.length
    && botEventsByInning[i] != null) {
    HalfInningResult r = simulateHalfInning(
        home.lineup, home.batterIndex, botEventsByInning[i], logExtraAfter3Out: true);
    home.batterIndex = r.nextBatterIndex;
    home.inningRuns[i] = r.summary.runs;
    home.totalRuns += r.summary.runs;
    show(r.summary);
} else {
    System.out.println("(本局無速記事件)");
}

```

執行結果

```

===== MLB 播報 =====
R=1
H=2
O=3
LOB=2 (三壘、二壘有人)
===== MLB 播報 =====
[2025-11-15 21:12:05] ERROR - 3 Out 之後仍出現速記事件：1B
R=0
H=0
O=3
LOB=0 (無人在壘)
===== MLB 播報 =====
[2025-11-15 21:12:05] ERROR - 無法解析速記代碼：WTF, 打者：Will Smith
R=1
H=1
O=1

```

```
LOB=1 (三壘有人)
===== MLB 播報 =====
R=4
H=1
O=2
LOB=2 (三壘、一壘有人)
```

```
===== 比賽開始 =====
```

```
----- 第 1 局上：客隊 Dodgers 進攻 -----
```

```
R=1
H=2
O=3
LOB=2 (三壘、二壘有人)
```

```
----- 第 1 局下：主隊 Blue Jays 進攻 -----
```

```
[2025-11-15 21:12:05] ERROR - 3 Out 之後仍出現速記事件：1B
```

```
R=0
H=0
O=3
LOB=0 (無人在壘)
```

```
----- 第 2 局上：客隊 Dodgers 進攻 -----
```

```
[2025-11-15 21:12:05] ERROR - 3 Out 之後仍出現速記事件：1B
```

```
R=0
H=0
O=3
LOB=0 (無人在壘)
```

```
----- 第 2 局下：主隊 Blue Jays 進攻 -----
```

```
[2025-11-15 21:12:05] ERROR - 無法解析速記代碼：WTF, 打者：Alejandro Kirk
```

```
R=1
H=1
O=1
LOB=1 (三壘有人)
```

```
----- 第 3 局上：客隊 Dodgers 進攻 -----
```

```
R=4
H=1
O=2
LOB=2 (三壘、一壘有人)
```

```
----- 第 3 局下：主隊 Blue Jays 進攻 -----  
R=1  
H=2  
O=3  
LOB=2 (三壘、二壘有人)  
  
===== 記分板 =====  
Inning : 1 2 3 | R  
Dodgers : 1 0 4 | 5  
Blue Jays : 0 1 1 | 2  
===== 比賽結束 =====
```

參考資料與使用工具及比例（包含 AI）：

上課講義、期中考程式、AI(大約 70%)