**COMP1917 Computing 1**

**Session 2, 2014**

**Project 1 - Tennis Scoring**

Due: Sunday 31 August, 11:59 pm   
Marks: 8% of final assessment

**Introduction**

For this project, you will be writing a program to compute the score for a tennis game between two teams A and B.

Your program should read, from standard input, a sequence of characters indicating the winner of each individual point, and print to standard output the current score at various points during the game.

**Note**: although it has been released in Week 2, we recommended that you begin this assignment in Week 3, because some of the material covered in Week 3 (functions and arrays) will be very useful in helping you to structure your program appropriately.

**Tennis Scoring**

A tennis match is divided into "sets", "games" and "points". Full details of the tennis scoring system can be found at<http://en.wikipedia.org/wiki/Tennis_score>.

To win a game, one team must score at least four points and at least two points more than the other team has scored during that game.

For historical reasons, scores within a game are reported according to the following convention, where the number of points won by the serving team is shown on the left hand side, those of the receiving team along the top row:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | **0** | **1** | **2** | **3** |
| **0** | Love-All | Love-15 | Love-30 | Love-40 |
| **1** | 15-Love | 15-All | 15-30 | 15-40 |
| **2** | 30-Love | 30-15 | 30-All | 30-40 |
| **3** | 40-Love | 40-15 | 40-30 | Deuce |

Once both teams have three or more points, the score is reported as:

* "Deuce" if the two teams have won the same number of points
* "Advantage Server" if the serving team is ahead by one point
* "Advantage Receiver" if the receiving team is ahead by one point

To win a set, one team must win at least 6 games, and at least 2 games more that the other team has won during that game.

To win the match, one team must win at least 3 sets.

**Note**: We will follow the rules used at Wimbledon, where a tie-break is never played in the 5th set. If the 5th set goes to a score of 6-6, normal games continue to be played until one team has a 2-game lead. Unless attempting the "Bonus" section, you can assume that the first four sets in the match will never go to a score of 6-6, so you do not have to worry about tie-breaks.

If Team A serves for the first game, then Team B will serve for the second game, and the serve will continue to alternate between the two teams for the rest of the match, without regard to sets; in other words, if Team A serves for the last game of one set, then Team B will serve for the first game of the following set.

**Input and Output**

The input to your program will be a series of characters A, B and S, with line breaks at appropriate intervals. The character A indicates a point won by Team A; B indicates a point won by Team B.

The character S indicates that your program should print the score at that point in the match, in the following format:

Team ? to serve:

?-? ?-? ?-?

???-???

The first line indicates which team is about to serve (A or B); the second line indicates the score for all completed sets as well as the current set, and the third line indicates the score within the current game (from the perspective of the serving team). For each set, the number of games won by the serving team should be printed first, followed by a dash, followed by the number of games won by the receiving team.

When your program detects that the match is finished, it should print the final score in this format:

Team ? wins: ?-? ?-? ?-? ?-?

The final score is a list of the scores for each set, with the score of the match-winning team printed first. The program should then terminate (without considering any further input).

For example, if the input to your program is BSAAAABBBS, the output should look like this:

Team A to serve:

0-0

Love-15

Team B to serve:

0-1

40-Love

This indicates that, when the first S instruction is processed, no sets or games have been won by either team, and Team B (who are receiving) have scored one point in the first game. When the second S is processed, Team A has won one game, and Team B (who are now serving) have scored three points in the second game. Note that a blank line should be printed under each score.

A compiled solution to the assignment will be provided for you in the [tools](http://www.cse.unsw.edu.au/~cs1917/14s2/hw1/tools) directory. Your program should produce **exactly** the same output as this executable.

**Submission**

You must submit **one** ANSI C source file which **must** be called hw1.c which, when compiled, will produce an executable that performs exactly as described in the specification. Once submissions are open, you should submit by typing

give cs1917 hw1 hw1.c

You can submit as many times as you like - later submissions will overwrite earlier ones. You can check that your submission has been received by using the following command:

1917 classrun -check

The submission deadline is Sunday 31 August, 11:59 pm.  
15% penalty will be applied to the (maximum) mark for every 24 hours late after the deadline.

Additional information may be found in the [FAQ](http://www.cse.unsw.edu.au/~cs1917/14s2/hw1/faq.shtml) and will be considered as part of the specification for the project. Questions relating to this project can also be posted to the MessageBoard on the course Web page.

If you have a question that has not already been answered on the FAQ or the MessageBoard, you can email it to your tutor, or tocs1917.hw1@cse.unsw.edu.au

**Marking scheme**

* 6 marks for correct programming and functionality
* 2 marks for style and comments

You should always adhere to good coding practices and style. Your program must conform to the [Style Guide](http://www.cse.unsw.edu.au/~cs1917/14s2/style_guide.html) for the course, and must be adequately commented. Programs that generate compilation errors will receive a very low mark, no matter what other virtues they may have. In general, a program that attempts a substantial part of the job but does that part correctly will receive more marks than one attempting to do the entire job but with many errors.

**Bonus challenge**

One bonus mark is available if you can modify your program so that it correctly handles **tie-breaks**. Details of how a tie-break is conducted can be found at <http://en.wikipedia.org/wiki/Tennis_score>. To win the tie-break, one team must score at least 7 points and at least 2 points more than the other team has scored during the tie-break. Scores during a tie-break should be printed as 0,1,2,3, etc. rather than Love,15,30,40, etc. For example, a score of 5-3 means that the team serving the first point of the tie-break has won 5 points; the other team 3 points. When printing the final score for a tie-break set, the number of points scored by the losing team should be given in brackets. For example: 6-7(8) means that the tie-break was won by the (currently) receiving team, and that the losing team scored 8 points (which means the winning team must have scored 10 points).

**Plagiarism Policy**

Your program must be entirely your own work. Plagiarism detection software will be used to compare all submissions pairwise (including submissions for similar assignments in previous years, if applicable) and serious penalties will be applied, particularly in the case of repeat offences.

**DO NOT COPY FROM OTHERS; DO NOT ALLOW ANYONE TO SEE YOUR CODE**

Please refer to the [Yellow Form](http://www.cse.unsw.edu.au/people/studentoffice/policies/yellowform.html), as well as the [CSE Addendum to the UNSW Plagiarism Policy](http://www.cse.unsw.edu.au/~chak/plagiarism) if you require further clarification on this matter.

Good luck!