Tennis Scoring

Tennis is a funny sport. To win a single tennis game you need to score 4 or more times against your opponent. However when they score they say 15 on the first point, 30 on the second, 40 on the third and they say "love" when you have not scored yet. If you have a score 40-15 it means that you have scored 3 times and your opponent only once. Love-30 means that you have yet to score and your opponent has scored twice.

To win a game you must be ahead by at least 2 so for example if the score is 40-40 (both scored 3 times), the next time you score would be your 4th time but you do not win yet because the difference to your opponents score is only 1 point. You are said to have the "Advantage" because if you score the next point you will win (adv-40). If your opponent scores, it goes back to 40-40 (called deuce) and the process repeats until a winner is found.

It gets better, tennis matches are not won in a single game, but a "set" of games. You must win at least 6 games to win a set but in the same way you must win by 2 or more games to win the set. You must win 2 sets to win the match (assume only 3 sets will be played maximum). If you are not clear on the rules, ask for clarification.

You have been asked to write a program to keep track of the score of a tennis match given a stream of input from a user watching the game. It starts with the two names of the players playing the match followed by the name of the player who scores each point. When the word "display" is read, your output will be the current score of the match in the following format:

```
playerName1  S G P
playerName2  S G P
```

where S is the number of sets won, G is the number of games win in the current set and P is the number of points in the current game. You may assume that the match is in progress (aka no one has won the match yet) and the display will only occur at the end of the file.

Two input files will be provided to you.

Input #1:	**Output #1:**
Federer Raonic Federer Federer Raonic display	Federer 0 0 30 Raonic 0 0 15

Input #2:	**Output #2:**
tennisData2.txt	Federer 0 1 love Raonic 0 0 30

Input #3:	**Output #3:**
tennisData2.txt	Federer 1 3 40 Raonic 0 0 love

Input #4:	**Output #4:**
tennisData4.txt	Federer 0 6 40 Raonic 0 7 Adv

Student Name:

Question - Assessment:	Application /8	Communication /6

Checklist for Peer/Self Evaluation of Programs: You will be give one full mark for each criterion in which you score either **Always** or **Excellent**. Marks will be deducted by ½ for every move to the left (as you approach **Never** or **Poor**).

Program Correctness (Application)

Criteria	Never	Some-times	Most of the time	Always	Comments
Design elements indicate correct idea for solution					
Correct program results					
Correct use of selection statements					
Correct use of repetition statements					
Correct use of subprograms and/or objects					

Programming Style (Communication)

Criteria	Poor	Fair	Good	Excellent	Comments
Documentation					
Proper subprogram and variables names					
Use of white space/indentation					
Program readability (Is the easy to read, follow and understand?)					