CM1103 Coursework 2016

Module:	CM1103 Problem Solving with Python
Lecturer:	Dr S M Allen
Date set:	Monday 21st November 2016
Submission date:	Friday 9th December 2016
Submission arrangements:	An electronic copy must be submitted (see *Deliverables* below) via Learning Central before 17:00
Title:	CM1103 coursework 2016
Indicative effort:	Less than 10 hours

This coursework is worth 20% of the total marks available for this module. The penalty for late or non-submission is an award of zero marks. You are reminded of the need to comply with Cardiff University's Student Guide to Academic Integrity. Your work should be submitted using the official Coursework Submission Cover sheet.

Instructions

This coursework asks you to simulate games of squash between players of different ability. Read the description of scoring methods on page 3, and answer the questions on page 4.

Submission instructions

You must submit via Learning Central:

Description	**Type**	**Name**
Cover sheet	One PDF (.pdf) file	[student number].pdf
All source code used to answer the questions	One or more Python source files (.py)	No restriction
A transcript of an interactive Python session that shows the results of the test cases given for each question.	One file saved as either text (.txt), Python (.py), PDF (.pdf) or Word (.doc or .docx)	No restriction
A document containing the figures generated for questions 1d and 2	One file saved as either PDF (.pdf) or Word (.doc or .docx)	No restriction

Note the School policy on electronic coursework submission `http://www.cs.cf.ac.uk/currentstudents/ElectronicCourseworkSubmissionPolicy.pdf`.

Criteria for assessment

Credit will be awarded against the following criteria

Code quality:

- Is the code elegant and well-written?
- Are functions written so that they can be reused between questions?
- Are appropriate control structures used?
- Is the code simplified by the use of built-in language features where appropriate?
- Is the code readable and easy to follow?
- For question 2, is the code concise and does it reuse other functions?

Functionality:

- Does the code perform the required task?
- Is the code free from errors?
- For question 2, are a suitable range of ability values and match lengths used to generate the figure?

Presentation:

- Are the figures clear and correctly formatted?

Feedback

Feedback on your coursework will address the above criteria.

Individual feedback will be returned by the end of Week 12 (i.e. after 12 University working days).

Group feedback will be provided in the revision lecture in week 12 and via model solutions directly after all student submissions have been made.

Rules of squash

This coursework uses a simplified summary of the rules of squash. Squash is a racquet game played by two players, and consists of a number of rallies. In each rally, the player who starts is the *server*, and the receiving player is the *returner*. A player wins a rally if the other player is unable to make a legal shot. Matches are usually played over a number of games, with each game decided by the player who reaches a certain score first.

There are two scoring systems commonly used in squash:

Point-a-rally scoring (PARS)

- The winner of each rally always receives a point (regardless of whether they were the server or returner).
- The first player to reach at least 11 points and be ahead by at least 2 points wins the game.
- If the server wins a rally, they continue as server.
- If the returner wins a rally, they become the server.

English scoring

- *Only the server is awarded a point if they win a rally.*
- If the server wins a rally, they receive a point and continue as server.
- If the returner wins a rally, they become the server but don't receive a point.
- The first player to reach 9 points wins the game **unless** the score has reached 8-8.
- If the score reaches 8-8, the player who reached 8 first decides whether to play to 9 or to 10.

Modelling playing ability

- Assume a player A's ability is represented by a value r_A such that $0 < r_A \leq 100$.
- In a game between player A and player B, the probability that A wins any given point is:

$$P(A \text{ wins}) = \frac{r_A}{r_A + r_B}$$

Questions

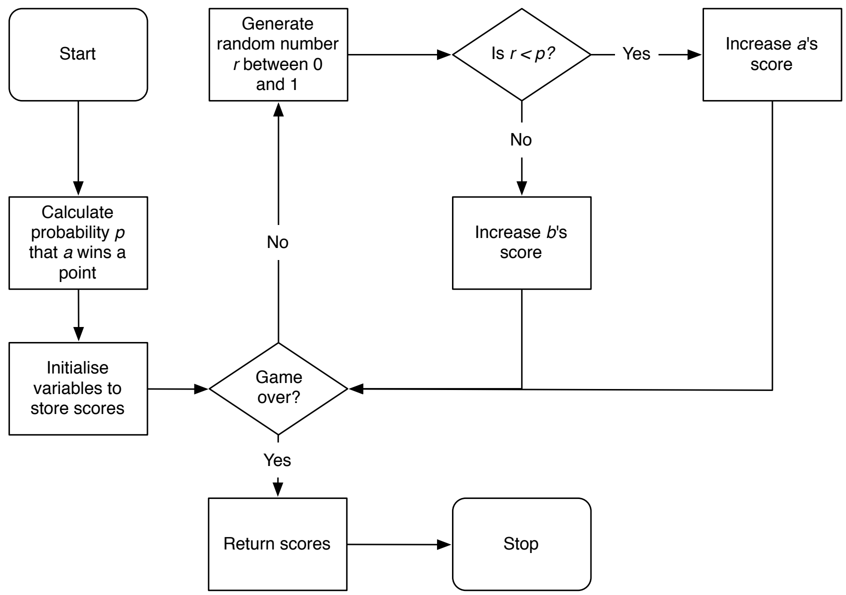
1. Provide code to answer the questions below. You should also provide a transcript of a Python interactive session that shows the result of executing the test case given for each question.

   There will be an additional mark covering the quality and style of your code.

   All parts of this question use the *Point-a-rally scoring (PARS)* system.

   [Code quality: 5 marks]

   (a) Define a function `game(ra, rb)` that implements the algorithm below to simulate a single game of squash between players with abilities r_a and r_b. The function should return the final score.



   If the random seed is set to `57`*, then calling the function with arguments* `70` *and* `30` *should return* `(11, 5)`.

   [Functionality: 6 marks]

   (b) Define a function `winProbability(ra, rb, n)` that simulates `n` games in order to estimate the probability that a player with ability r_a will win a game against a player of ability r_b.

   Given sufficient simulations, calling the function with abilities `70` *and* `30` *should give the answer* `0.98` *when printed to 2 decimal places.*

   [Functionality: 3 marks]

   (c) Assume players abilities are given in a csv file with the format:

```
player a ability, player b ability,
60, 20,
100, 55,
50, 40,
20, 70,
95, 85,
```

Write a function that reads in a csv file of this format and returns a `list` of `tuple`s containing each pair of abilities.

Reading in the data above should return: `[(60, 20), (100, 55), (50, 40), (20, 70), (95, 85)]`.

[Functionality: 3 marks]

(d) Write a function that takes a `list` of the format returned by your answer to 1c as an argument, and uses `matplotlib` to produce a plot showing the probability that player a beats player b (in a game) against r_a/r_b for each pair.

Show the figure produced for data `[(60, 20), (100, 55), (50, 40), (20, 70), (95, 85)]`

[Presentation: 2 marks; Functionality: 3 marks]

(e) Suppose player *a* has ability 60 and player *b* has ability 40, and they play a match where the winner is the first player to win n games. What is the smallest value of n such that the probability that *a* wins the match is at least 0.9?

You may answer using simulation, theory, or a combination of both.

[3 marks]

2. Assume each ~~point~~ rally in a game takes equal time. Plot a graph that shows how the length of a match varies between the two scoring systems (English and Point-a-rally scoring (PARS)) depending on the relative ability of the two players. Clearly state any assumptions you make.

[Presentation: 3; Functionality: 7 marks; Code quality: 5 marks]