CM1103 Coursework 2017

Module:	CM1103 Problem Solving with Python
Lecturer:	Prof S M Allen
Date set:	20th November 2017
Submission date:	9:30am on 15th December 2017
Submission arrangements:	An electronic copy must be submitted (see *Deliverables* below) via Learning Central before 9:30 am
Title:	CM1103 coursework 2017
Indicative effort:	Less than 15 hours

This coursework is worth 40% 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.

Problem definition

A small, local branch of a bank employs a single teller. The manager is considering ways to improve customer satisfaction by reducing queuing times. This coursework will use computer simulation to provide the necessary evidence to support their decision.

Deliverables

You must submit:

1. A single source code file based on the template provided for Question 1. This file should also contain your code to answer Question 2.
2. A pdf file with the figure to answer Question 2.
3. A Word file (based on the template provided) of no more than 2 pages that answers Question 3, and any source code used (this can be combined with your answer to 1 if desired).

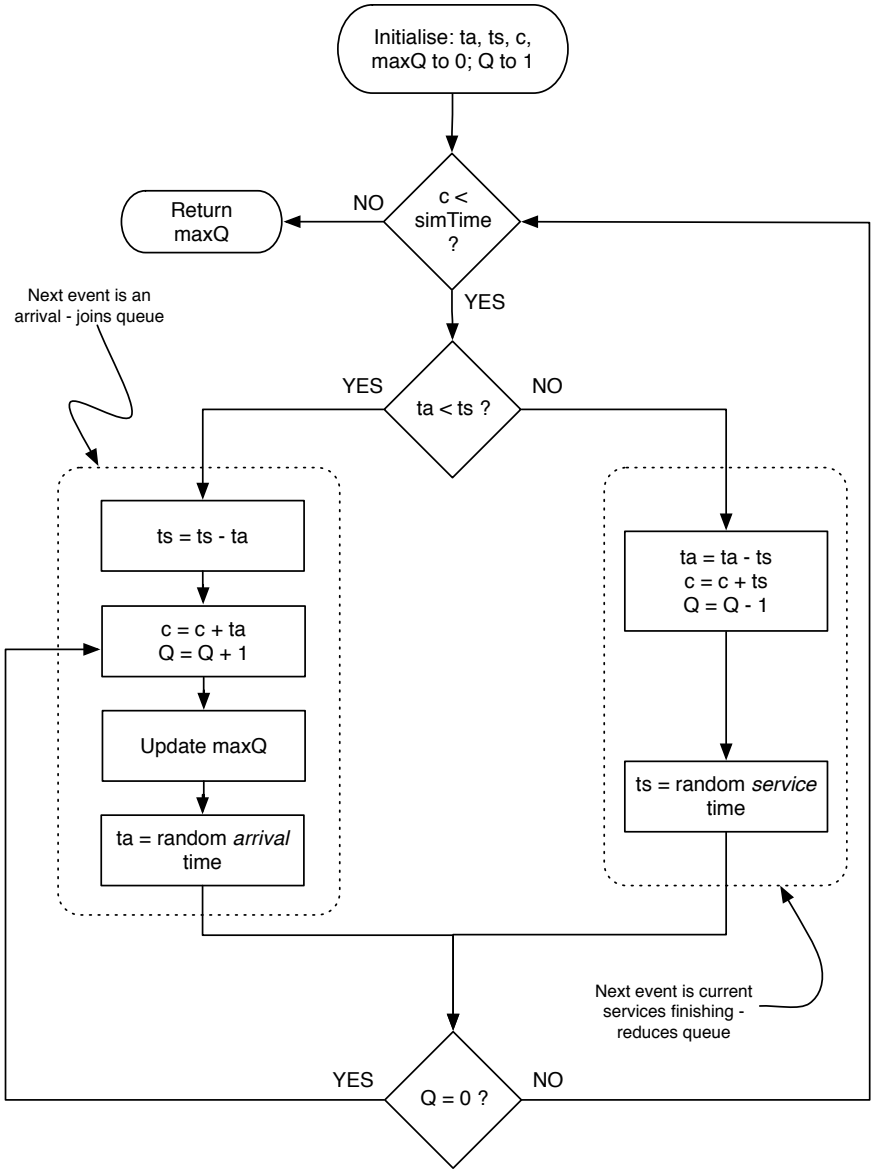
Description	**Type**	**Name**
Cover sheet	One pdf file	`[Student number].pdf`
Answer to Q2	One pdf file	`Figure_[Student number].pdf`
Answer to Q3	One Word or pdf file	`Report_[Student number].pdf`
Source code for Q1, 2 and 3	One or more Python files (may be contained in a single zip file)	No restriction

Simulating a single queue

1. The key parameters in this simulation are the rate at which customers arrive, and the length of time each customer takes with the teller.
2. Let α be the mean gap between customers arriving.
3. Let β be the mean time that a teller will take to serve a customer.
4. A function `nextTime` is provided for you that will generate appropriate random times when passed a desired mean.
5. The following algorithm can be used to simulate a single queue over a period of `simTime` minutes. It uses variables:

   - `c` to store the current time (measured in minutes) within the simulation
   - `ta` to store the time to the next customer arrival
   - `ts` to store the time until the current customer has finished being served
   - `Q` to store the current length of the queue
   - `maxQ` to store the size of the longest queue

   The algorithm progresses by repeatedly incrementing the current time to the time of the next critical event – either a customer arriving, or a customer leaving after their service has finished.


Initialise: ta, ts, c, maxQ to 0; Q to 1
c < simTime ?
NO
Return maxQ
YES
Next event is an arrival - joins queue
YES
ta < ts ?
NO
ts = ts - ta
c = c + ta
Q = Q + 1
Update maxQ
ta = random arrival time
ta = ta - ts
c = c + ts
Q = Q - 1
ts = random service time
Next event is current services finishing - reduces queue
YES
Q = 0 ?
NO

Questions

1. Complete the Python source code template provided to answer the questions below. You will be given a *functionality* mark for each question that will be largely based on how well your code satisfies a range of doctests (some of these are provided for you – others may also be used).

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

   - Is the code readable and easy to follow?
   - Is the code elegant and well-written?
   - Is the code simplified by the use of built-in language features where appropriate?

   [Code quality: 5 marks]

   (a) Assume the time between customer arrivals is generated by the function `nextTime` with mean α, and the time taken to serve a customer is generated by the function `nextTime` with mean β. Mathematical theory then states that (over a suitably long period) the mean queue length is equal to:

   $$\frac{\frac{\beta}{\alpha}}{1-\frac{\beta}{\alpha}}$$

   Complete the function `theoreticalMeanQueueLength` so that it calculates this value given parameters α and β.

   [Functionality: 3 marks]

   (b) Complete the function `checkMean` so that it calls the function `nextTime` a specified number of times for a desired mean, and calculates the actual mean of the random times returned.

   [Functionality: 3 marks]

   (c) Suppose you are required to perform simulations for a number of scenarios whose parameters are given in a csv file with the format:

   ```
   alpha (m), beta (m), simTime, time unit,
   10, 2, 480,,
   5, 3, 480,,
   20, 15, 240,,
   20, 2, 8, h,
   15, 2.5, 360,,
   ```

   where the first column specifies the mean customer arrival time; the second specifies the mean service time; and column three specifies the simulation time in minutes **unless** there is a `h` in column 4 – in this case the value in column 3 contains a number of hours.

   Complete the function ~~`readcsv`~~ `readExperimentParameters` so that it reads in a csv file of this format and returns a `list` of `tuples` with 3 elements containing the parameter sets **with all times given in minutes**.

   [Functionality: 3 marks]

   (d) Complete the function ~~`simulate`~~ `singleQueue` so that it implements the queuing algorithm above to simulate a queue with the specified parameters, and returns the **maximum** queue length.

   [Functionality: 6 marks]

2. Using `matplotlib`, plot a line graph that shows how the theoretical mean queue length varies as α varies between 1.1 and 10 using steps of 0.1 while keeping $\beta = 1$ ~~$\frac{\beta}{\alpha}$ varies between 1.1 and 10 using steps of 0.1~~ . Provide your source code and a pdf file containing the figure, remembering to label the axes and provide a title.

   [Functionality: 5 marks]

3. Produce a scientific report using the supplied template, that demonstrates ~~compares~~ the customer service benefits that would arise from:

   (a) Purchasing new equipment that reduces the mean service time per customer (your answer should consider a range of different percentage reductions).

   (b) Employing a second teller.

   Your conclusions should be based on an appropriate number of simulations for an appropriate range of experimental parameters, and consider average vs worst case customer experience.

   If you modify the algorithm, you **do not** need to include the modified pseudocode in your report.

   [Report: 6 marks; Acheivement: 4 marks; Code quality: 5 – see template for more detail]

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 to question 1 directly after all student submissions have been made.