Students are encouraged to browse through chapter 4 of the textbook. A good chunk of it will hopefully review. For this particular LGA, **students should read** §4.1.2 (particularly the definitions and theorem, you may ignore the proof) and §4.1.4 (the definitions and theorems around *sample path* statistics).

The following numbered questions should be split across your group and the solutions discussed during the next lecture period. Students should review the learning goals for the day, determine which are applicable to their questions and provide answers or commentary to their group members. When using the Internet to formulate answers (some questions may require this), keep track of **where** you find your information on the web. You may be asked for, and are expected to have (in Email-able form), URLs supporting your investigations.

1. Welford's equations are an iterative approach to calculating $\bar{x}, s$ described in § 4.1.2 of the text.

   (a) Use the textbook's definition 4.1.2 and theorem 4.1.2 (page 139) to design an API or object that is initialized with $\bar{x} = 0$ and $v_0 = 0$, permits data points to be added dynamically, and can return the values $n = i$, $\bar{x}_i$ and $s = \sqrt{v_i/i}$ using a standard interface.

   (b) Now, incorporate your shiny new code into a simple application that accepts a single command line parameter which will be a file of floating point values. The file will have one value per line. Your program should output at least $\bar{x}$ and s for the data. Display all results **with eight decimal places of precision**. An example run might look like:

   ```
   $ ./welford dataset1.dat
   8.50254480          1.35171337
   ```

   Run your code on these data files: `dataset1.dat` , `dataset2.dat` and `dataset3.dat` and provide the results to your group when you gather again in lecture.

   Be prepared to review your object or API design with your group when you meet again.

2. Recall that integration of continuous functions in *discrete event simulations* are as easy as adding up rectangles (because variables change values only during the processing of events at a distinct simulation time). § 4.1.4 of the textbook describes the set of Welford's equations for maintaining $\bar{x}$ and s from a time-domain **sample-path** of a function. Use the equations of theorem 4.1.4 (page 146) and definition 4.1.4 to implement an API or object as in the previous question 1. (δ_i in theorem 4.1.3 is described on the preceding page.) Note that in this case each new data point is a tuple (t_i, x_i).

   Slap this new code inside a small application for testing. The $\bar{x}$ and s for the sample path in `samplepath.dat` are $\bar{x} = 2.77217$ and $s = 2.69584$.

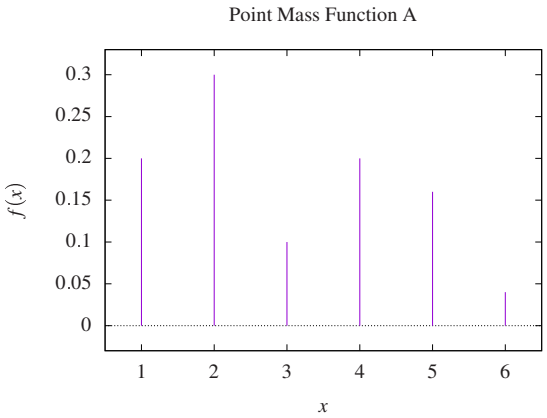
   Be prepared to review your object or API design with your group when you meet again.

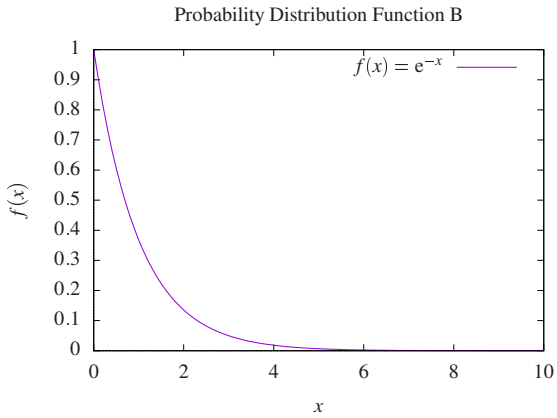
3. For all of the following questions, be sure to provide explanations for your answers, **do not** simply cite numerical answers. When your group reconvenes next lecture be prepared to quickly walk through the logic for each answer's calculation.

   (a) i. (5 points) The pdf $f(x) = (x-2)^2$ exists on $1 \leq x < b$. What is b to three decimal places?

   ii. (5 points) Plot the CDF (or $F(x)$) of $f(x)$.

   (b) Given the following pmf and pdf (A and B below), complete the table and the remaining question.





i.

	A	**B**
$P(x=2)$	(a)	(b)
$P(1<x<2)$	(c)	(d)
$P(x=1 \text{ OR } x=2)$	(e)	(f)

   ii. What are the mean ($\bar{x}$) and sample standard deviation (s) for **A**?

   (c) The following is a table of rolled values for a bias 8-sided die after many rolls:

Face	1	2	3	4	5	6	7	8
Count	33	10	41	71	12	18	7	18

   i. Plot the frequency histogram for the data set.

   ii. Plot the (empirical) cumulative distribution function (cdf) for the data set.

4. There are two well known equations for calculating the sample variance or sample standard deviation (s^2 and s respectively, $s = \sqrt{s^2}$):

$$s^2 = \frac{1}{n}\sum_{i=1}^{n}(\bar{x} - x_i)^2 \tag{1}$$

and

$$s^2 = \left(\frac{1}{n}\sum_{i=1}^{n}x_i^2\right) - \bar{x}^2 \tag{2}$$

Note that implementing equation 1 in computerese requires two passes over the data (one to calculate $\bar{x}$, another to sum the squared deviations from $\bar{x}$). Whereas equation 2 can be implemented in one pass: maintain three variables `n`, `sum_xi_2` and `sum_xi`; when all the data has been processed calculating s is straightforward.

Write a program that accepts a single command line parameter as the argument. This parameter will be a data file holding textual floating point data, one data value per line. There will be a maximum of 2000 data points in a file.

Your program will use your own implementation of equations 1 and 2 to print the two calculated values for $\bar{x}$ and s from the given data file provided on the command line. Display all results **with eight decimal places of precision**. An example run might look like:

```
$ ./SIM dataset1.dat
two-pass          8.56097073          1.38878132
one-pass          8.56097073          1.38878132
```

For this assignment you **may not** rely on a language or library's statistical functions. You need to "write the loops" and make the calculations on your own. Use IEEE 8-byte **"double"** variable types to store all floating point values. These are `double` in C++ and Java, `float` in Python2 or Python3.

Test your code on these data files: `dataset1.dat` , `dataset2.dat` and `dataset3.dat` . Summarize the results of these calculations for your group.