1. Give and briefly describe one example (not listed in the lecture notes) of a process that you imagine we could model with Monte-Carlo methods.

2. Imagine that your commute into work involves 5 minutes of actual driving plus passing through 6 traffic lights. We’ll assume that for each traffic light there is a 50 - 50 chance that you will have to stop, and if you have to stop you will wait anywhere between 0 and 2 minutes (uniformly distributed). Write a (Monte-Carlo) function `timeToGetToWork()` that returns the time it takes to get to work on a particular morning.
3. Using the function `timeToGetToWork()` from problem 2 as a helper function, write a (Monte-Carlo) function `lateForWork(T)` that returns `True` if you are late for work and `False` if you are not, given that you leave `T` minutes before start time.

4. Write the Python statements (using `hist.py` module described in “Monte-Carlo Methods” notes) that are required to plot a histogram of the results of 10000 calls to function `timeToGetToWork()`. You should set `binMin` and `binMax` based on the shortest and longest possible commute times (what are these?).
5. Write a function `printMultTable(N)` that prints all integer pair products, where each integer is between 1 and N. For example, `printMultTable(3)` should print the following:

```
1 * 1 = 1
1 * 2 = 2
1 * 3 = 3
2 * 1 = 2
2 * 2 = 4
2 * 3 = 6
3 * 1 = 3
3 * 2 = 6
3 * 3 = 9
```

You will want to make use of a *nested for-loop* (a for-loop within a for-loop) to form all product pairs. See end of section 5.2 and section 5.3 of Perkovic.