Problem Wk.10.1.2: Conditional distributions

Read Section 7.2 of the course notes if you haven't already done so.
It is important to be able to represent conditional distributions, of the form *P(B | A)*. Conditional distributions are not, themselves, distributions. **We will represent conditional distributions as functions from values that the variable *A* can take on, to distributions over *B*.**

So, if `bar` is a conditional distribution representing *P(B | A)*, then `bar(a)` will be the distribution over *B* corresponding to *P(B | A = a)*.

For example, consider a situation where we have the variable `Year` with domain `(1,2,3,4)` and the variable `Grade` with domain `('a', 'b', 'c', 'd', 'f')`. We can represent the conditional distribution `P(Grade | Year)` as a function, called `PGgY`, which, given a value for `Year`, returns a `DDist` over the grades:

```
>>> PGgY(1)
DDist(a: 0.3, b: 0.3, c: 0.3, d: 0.07, f: 0.03)
>>> PGgY(2)
DDist(a: 0.25, b: 0.35, c: 0.3, d: 0.07, f: 0.03)
```

Let variable `Disease` have domain `('disease', 'noDisease')` and let variable `Test` have domain `('posTest', 'negTest')`.

Define a conditional distribution (a function) that takes a value of `Disease` and returns a `DDist` that represents the distribution of `Test`, conditioned on `Disease` having the specified value. The function should encode the facts that:

- *P(posTest | disease) = 0.98*, and that
- *P(posTest | noDisease) = 0.05*.

Think carefully about the distributions that are consistent with these facts.

For example: `PTgD('disease').prob('posTest')` should evaluate to `0.98`.

This is happening in the `dist` module, so you do not need to use `dist.DDist`, it's enough to type `DDist`.

```
def PTgD(diseaseValue):
    pass
```

6.01SC Introduction to Electrical Engineering and Computer Science

Spring 2011