This Python notebook builds a simple Bayesian genome-inference program. In this project, the genome is a single string of 0’s and 1’s that we are trying to infer from noisy data. This data consists of a number of *reads*. A read is a string of 0’s and 1’s that is generated by selecting a substring of the genome that will be measured, copying this substring, and then introducing errors into the resulting string by transforming 0’s into 1’s and vice versa. For example:

```
genome:                01001010101001101011
read without errors:   0010101010011
errors:                 *   *  *
final read:            0110111000011
```

Note that reads can overlap, and therefore some positions in the genome may be ‘covered’ by many reads. Other positions may be covered by zero reads. In this exercise, the step of determining from where in the genome each read originates (the alignment step) has already been performed for you. The reads are provided in an example file, which is a tab-delimited file with two columns, and one line for each read. The first column contains the start-position of the read (the 0-based position in the genome string that corresponds to the first position in the read string). The second column contains the read string itself. You should not assume that all of the reads are the same length, although the maximum read length is 1000. The example is sorted by the first column (the start-position). In practice, both the length of the genome string and the number of reads may be very large (billions). Note that there are no ‘insertion’ or ‘deletion' errors.

For example, the following situation:

```
genome position (tens):     00000000001111111111
genome position (ones):     01234567890123456789
genome (unobserved):        01001010101001101011
read 1:                      1001110101000
read 2:                         1010111001111
read 3:                         10101
read 4:                                  1101011
```

would be represented by the following example:

```
1       1001110101000
4       1010111001111
4       10101
13      1101011
```

The reads have error rates p01 and p10, which are the probability that a 0 becomes a 1 and the probability that a 1 becomes a 0, respectively, when the reads are generated. The errors are independent. You can assume that the genome is a random string of 0’s and 1’s, where each position is sampled independently, with probability p1 of being a 1. You can assume that the start-positions provided in the example file are correct. In summary, the parameters are:

p01	Probability that a 0 base is measured as a 1
p10	Probability that a 1 base is measured as a 0
p1	Prior probability that any given genomic base is a 1

This program takes an example file as input and produces a tab-delimited file with the posterior probability that the genome character is a 1 at each position, with one position per line. The position is the first column, and the probability is the second column. All positions in the genome from 0 through the last position measured in any read are included in the output (even if there are no measurements of the position). The parameters p01, p10, and p1 are arguments to the program, and you can assume these are correct (i.e. these parameters do not need to be estimated from the data). The probabilities are rounded to three decimal places.

Continuing with the previous example, the output with p01 = p10 = 0.10 and p1 = 0.5 will be:

```
0	0.500
1	0.900
2	0.100
3	0.100
4	0.999
5	0.100
6	0.999
7	0.001
8	0.999
...
18	0.900
19	0.900
```