CSCI 561: Foundations of Artificial Intelligence
Instructor: Sheila Tejada
Homework #3: Bayesian Inference
Due Date: July 31, 11:59 pm

Problem:

For this homework you will write a python program **“bayes.py”** that allows you to do inference in a Bayesian network. For this homework, you don’t need to worry about decision nodes or utility nodes. Your task would be to use variable elimination or enumeration to calculate joint probabilities, marginal probabilities, or conditional probabilities.

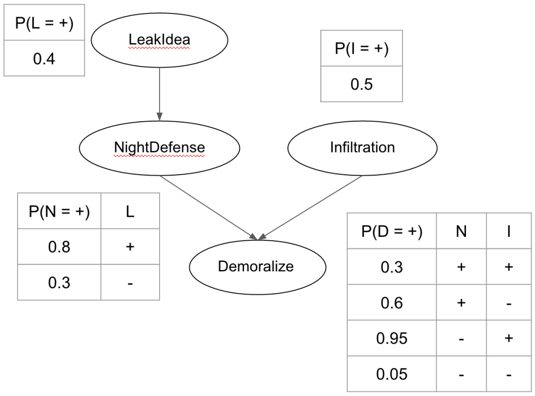
Pseudocode:

1. The enumeration algorithm for answering queries on Bayesian networks: AIMA Figure 14.9
2. The variable elimination algorithm for inference in Bayesian networks: AIMA Figure 14.11

Input format:

Sample:

```
2
P(NightDefense = +, Infiltration = -)
P(Demoralize = + | LeakIdea = +, Infiltration = +)
LeakIdea
0.4
***
NightDefense | LeakIdea
0.8 +
0.3 -
***
Infiltration
0.5
***
Demoralize | NightDefense Infiltration
0.3 + +
0.6 + -
0.95 - +
0.05 - -
```



P(L = +)
0.4

P(I = +)
0.5

P(N = +)	L
0.8	+
0.3	-

P(D = +)	N	I
0.3	+	+
0.6	+	-
0.95	-	+
0.05	-	-

You will be given a Bayesian network and several queries in a text file ending with a **.txt** extension which you will read through command line. The first line of the file will provide a number ’n’ which will be the number of queries in the file. The next ’n’ subsequent lines will have one query per line. After this, the file will provide the details about the Bayesian network and its probabilities in the format explained below. As shown in the sample file above, all nodes only have two possible values: “+” (event occurred) or “-” (event not occurred).

The input format is as follows:

- Each query will be asking for specific joint probabilities, marginal probabilities, or conditional probabilities.
- The tables of the Bayesian network are separated by three “*”, and will have the following format:

  e.g.
  ```
  Demoralize | NightDefense Infiltration
  0.3 + +
  0.6 + -
  0.95 - +
  0.05 - -
  ```
- The first line contains the node’s name, followed by the name of its parents, separated by a “|” sign (or no “|” sign when there’s no parent).
- All node names begin with an uppercase letter and contains letters only.
- The following lines show the probabilities of all combinations of parent node values.
- **All nodes can have only two possible values, “+” (event occurred) or “-” (event not occurred).**
- The probability will range from 0 to 1 (but not exactly 0 or 1), and is always for occurrence “+” only. Your program should compute the probability for nonoccurrence “-”.

  For example, given
  ```
  P(LeakIdea = +) = 0.4
  ```
  in sample file, your program should compute:
  ```
  P(LeakIdea = -) = 1 - P(LeakIdea = +) = 0.6
  ```
- The parent node values always follow the order in which they appear in the first line.
- With the exception of the first line, there is no specific order between lines of the tables (e.g. “+ -” may appear after “- +”).
- You may assume each node may have at most 3 parent nodes (8 lines of probabilities).
- When a node has no parent, then there is only a single number (probability of occurrence):

  e.g.
  ```
  LeakIdea
  0.4
  ```
- Every node has a corresponding table in the input file, so you can know the network structure from the first line of each table. For example, your program should figure out the network structure in sample file based on the following information:
  ```
  LeakIdea
  NightDefense | LeakIdea
  Infiltration
  Demoralize | NightDefense Infiltration
  ```
- There won't be any directed cycles in the given networks.
- Parent nodes always have their tables appearing before the child node.

Notes:

- All node names (such as `LikeIdea`) are case-sensitive alphabetical strings that begin with uppercase letters.
- All event names, numeric values and signs are separated by whitespace, except for query function "P", parentheses and the comma sign (",") in a query. Please follow the format of the sample files carefully when parsing the input file.
- You can assume that the input format is exactly as it is described. There are no errors in the given input.
- There will be at most 10 queries per input file.

Output format:

The result should be printed to a file called output.txt. Given the sample input above, the output content should be as follows:

Sample:

```
0.25
0.43
```

For each query in the input file, your program should generate a corresponding result (one result per line) as the output. The result for each query will be a decimal value between 0 and 1, rounded to two decimals. **Don't** print additional whitespace after the value, or extra line break in the end.

Notes:

- **You will get zero points if you don't follow the output format exactly.** Any regrading requests about this will be ignored.
- With the given description, we don't believe that multiple outputs are possible for any test case. If you are able to think of any such case, please let us know and we will make the necessary changes in the grading guidelines.
- The final test cases will be different from the sample test cases provided. Your assignment will be graded based on the performance on the final test cases only.

Guidelines:

- You can use Python2.7 or Python3.4 to implement your code. Make sure that the code runs on Vocareum.
- The name of the code should be **"bayes.py"** for Python 2.7 and **"bayes3.py"** for Python3.4.
- The command to run your code would be "python bayes{3}.py *inputfile*", where you will read the input file name from the command line argument.
- Input file is a text file ending with ".txt" extension.
- **Late submissions:** Late submissions will be allowed for additional 24 hours, i.e. till **August 1, 11:59 pm**. The late submission **penalty will be 50%** of the score you get. Homework will not be accepted after August 1.
- Homework-3 is worth 15% of the total grade for the course.

- Multiple submissions are allowed, and your last submission will be graded. So you are encouraged to submit early and often in order to iron out any problems, especially issues with the format of the final output. The performance of you program will be measured automatically; failure to format your output correctly may result in very low scores, which will not be changed.
- This is an individual assignment. You may not work in teams or collaborate with other students. You must be the sole author of 100% of the code you turn in.
- You may not look for solutions on the web, or use code you find online or anywhere else.
- You may use external resources to learn basic functions of Python (such as reading and writing files, handling text strings, and basic math), but the computation performed by the program must be your own work.
- Failure to follow the above rules is considered a violation of academic integrity, and is grounds for failure of the assignment, or in serious cases failure of the course.