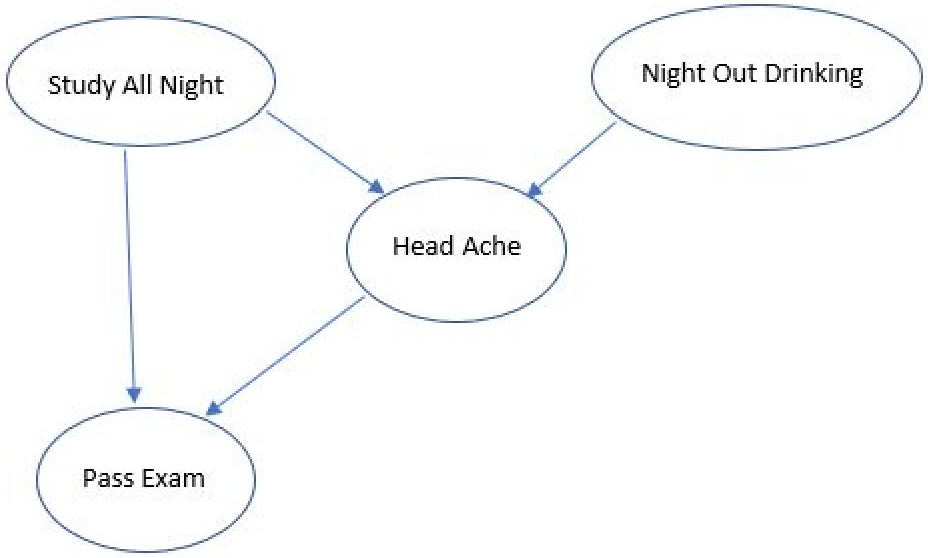
Our network example is based on a student final exams period, he or she will party, study, drink and because of that actions we want to know what are the chances of the person to pass the final exams.



Night Out Drinking	
T	F
0.05	0.95

Study All Night	
T	F
0.22	0.88

		Headache	
Study All Night	Night Out Drinking	T	F
T	T	0.91	0.09
T	F	0.63	0.27
F	T	0.85	0.15

F	F	0.001	0.999

		Pass Exam	
Study All Night	Head Acke	T	F
T	T	0.71	0.29
T	F	0.95	0.05
F	T	0.35	0.65
F	F	0.52	0.48

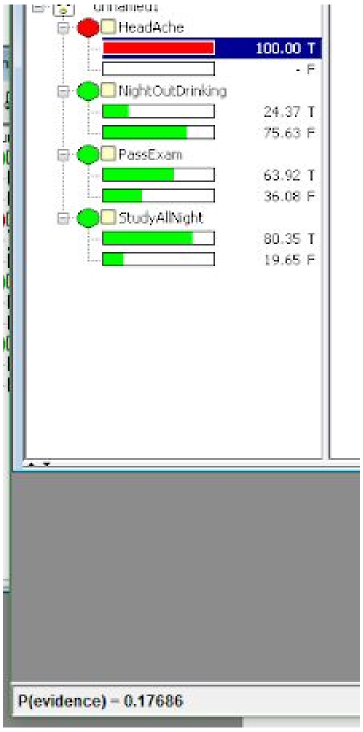
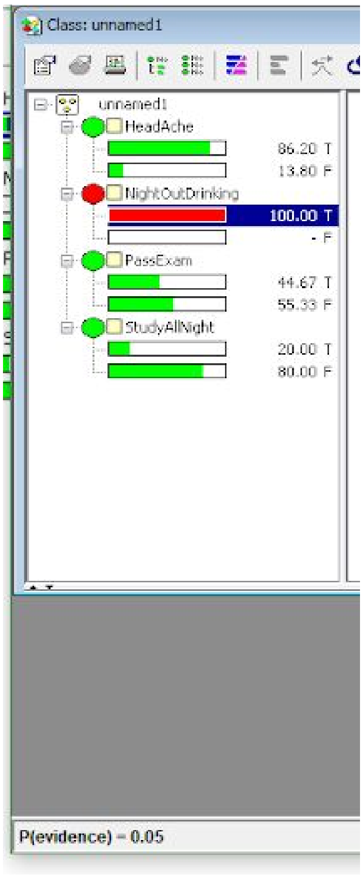
Encode your example into the input format for your program

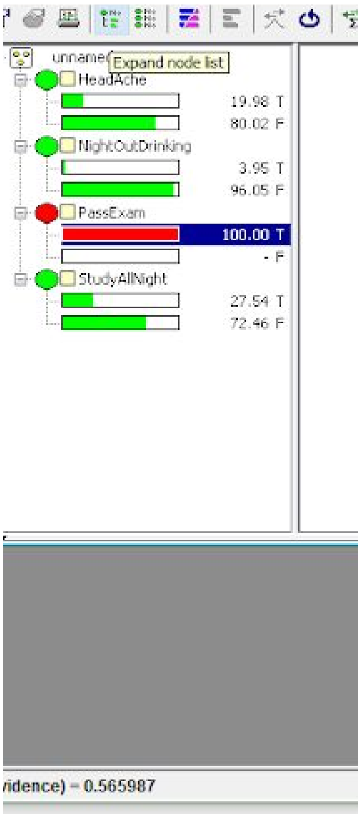
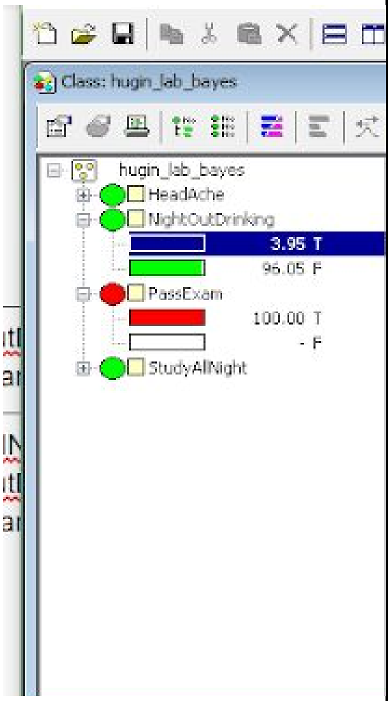
```
StudyAllNight, NightOutDrinking, HeadAche, PassExam
10
+NightOutDrinking=0.05
+StudyAllNight=0.22
+HeadAche|+StudyAllNight,+NightOutDrinking=0.91
+HeadAche|+StudyAllNight,-NightOutDrinking=0.63
+HeadAche|-StudyAllNight,+NightOutDrinking=0.85
+HeadAche|-StudyAllNight,-NightOutDrinking=0.001
+PassExam|+StudyAllNight,+HeadAche=0.71
+PassExam|+StudyAllNight,-HeadAche=0.95
+PassExam|-StudyAllNight,+HeadAche=0.35
+PassExam|-StudyAllNight,-HeadAche=0.52
5
+HeadAche
+NightOutDrinking
+PassExam
+NightOutDrinking|+PassExam
+StudyAllNight,+NightOutDrinking|+PassExam
```

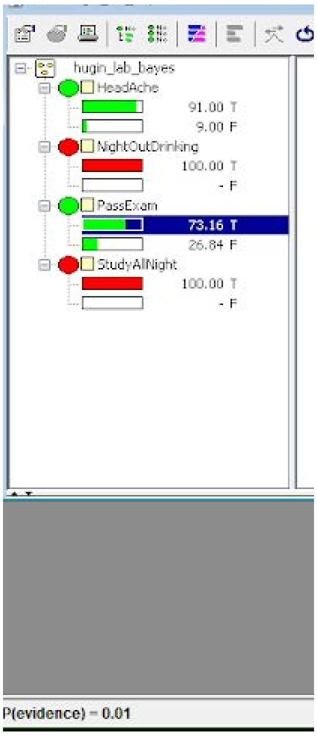
Querys:

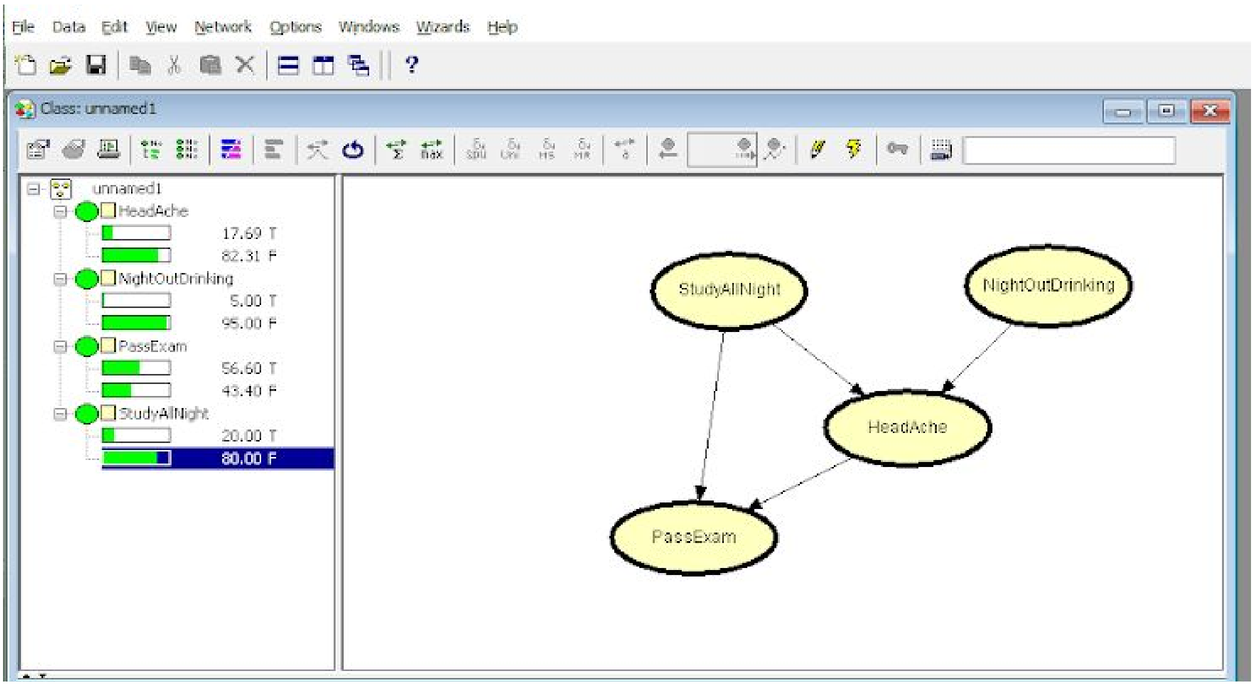
```
+HeadAche
+NightOutDrinking
+PassExam
+NightOutDrinking|+PassExam
+StudyAllNight,+NightOutDrinking|+PassExam
```

Compare the output to Hugin Lite to see that the values between both match.

Query	Our program	Hugin
+HeadAche	0.175571	 unnamed1 HeadAche 100.00 T - F NightOutDrinking 24.37 T 75.63 F PassExam 63.92 T 36.08 F StudyAllNight 80.35 T 19.65 F P(evidence) = 0.17686
+NightOutDrinking	0.05	 Class: unnamed1 unnamed1 HeadAche 86.20 T 13.80 F NightOutDrinking 100.00 T - F PassExam 44.67 T 55.33 F StudyAllNight 20.00 T 80.00 F P(evidence) = 0.05

+PassExam	0.5748353	 HeadAche 19.98 T 80.02 F NightOutDrinking 3.95 T 96.05 F PassExam 100.00 T - F StudyAllNight 27.54 T 72.46 F vidence) = 0.565987
+NightOutDrinking | +PassExam	0.0394758	 hugin_lab_bayes HeadAche NightOutDrinking 3.95 T 96.05 F PassExam 100.00 T - F StudyAllNight

+StudyAllNight, +NightOutDrinking	0.011	 hugin_lab_bayes HeadAche 91.00 T 9.00 F NightOutDrinking 100.00 T - F PassExam 73.16 T 26.84 F StudyAllNight 100.00 T - F P(evidence) = 0.01



unnamed1
HeadAche
17.69 T
82.31 F
NightOutDrinking
5.00 T
95.00 F
PassExam
56.60 T
43.40 F
StudyAllNight
20.00 T
80.00 F

Write a reflection or make a table where you compare Hugin Lite to your implementation.

What are the differences between what they generate?

The programs generate some equal values and some quite similar, the operations might be different. The way to show the data also is different, we normalize between 0 and 1 and hugin gives values in the range of 0 to 100.

Do they use the same algorithms?
They both use as base the Bayesian network theory. But we don't think they use the same algorithm, our algorithm is like a hard coded version, and the Hugin version it's obviously a better Order than ours.

What are their common bases?
The common base is the use of Bayesian theory in all calculations, they might use them in a different way but we guarantee that they are based on the Bayes Network theory.

Which tool would you use for what cases in real life applications? (400 to 500 words)

In the scenario of teaching to someone or show how the bayesian networks behave; The best option is to use the Hugin software. The first difference between our program and Hugin software as all may noticed is the lack of a graphical interface, this is very useful to show how the bayesian network looks, connects and interacts.
Also in a scenario where the bayesian network is in constant change, it would be better to use the Hugin software because there you can easily manipulate, add, or remove nodes and probabilities. Although Hugin software is not simple to understand and master, our program is limited by just working in giving the probabilities with specific inputs.
If a person just need to know specific probabilities given some input, then our program is the best option, like the probability of having a car crash, if its raining or n scenarios.
Both are good, but overall maybe the Hugin software dominates over our program by the simple fact that it was created by specialists and it is a more robust tool. On the other hand, If the user is trying to solve a simple and structured problem were such robustness is not needed, it would be recommended to use out tool for a couple of reasons: it is really light and easy to setup (in contrast with the Hugin software), it is easy to understand and to use, there are just a few set of rules to configure the input file and are quite obvious, and the output is just the set of probabilities that the user is looking for, with no more information than the one requested.