UNIVERSITY OF
LINCOLN

Lincoln School of Computer Science

Assessment Item Briefing Document

Title: CMP9132M Advanced Artificial Intelligence **Assessment Item 1 (assignment)**	**Indicative Weighting: 50%**

Learning Outcomes:

On successful completion of this assessment item a student will have demonstrated competence in the following areas:

- [LO1] Critically appraise a range of AI techniques for decision-making, problem solving and learning, identifying their strengths and weaknesses, and selecting appropriate methods to serve particular roles;
- [LO3] Design and develop an AI-based software program for solving complex search problems in an application domain of interest.

Requirements

This assessment comprises two assessed Tasks, as detailed in the following page.

1. TASKS ON PROBABILITIES (Task 1.A and Task 1.B). Weighting: 40% of this component.

2. TASK ON HIDDEN MARKOV MODELS. Weighting: 60% of this component.

Your submission should include a concise report (maximum 5 pages, not including the cover sheet and appendixes) that describes your work on the above tasks.

Useful Information

This assessment is an individual assessment component. Your work must be presented according to the Lincoln School of Computer Science guidelines for the presentation of assessed written work. Please make sure you have a clear understanding of the grading principles for this component as detailed in the accompanying Criterion Reference Grid.
If you are unsure about any aspect of this assessment component, please seek the advice of a member of the delivery team.

Submission Instructions

The deadline for submission of this work is included in the School Submission dates on Blackboard.

You must make an electronic submission of your work in PDF format by using the assessment link on Blackboard for this component. Each student is then required to discuss their solution to the module instructors, as per the schedule indicated on Blackboard. You must attend the lectures for further details, guidance and clarifications regarding these instructions.

DO NOT include this briefing document with your submission.

1. TASKS ON PROBABILITY

For each one of the following tasks, you should explain in the report the different steps taken to develop the requested software. This will be used by the marker to check inconsistencies or unclear points about the software. The source code must be included in an appendix at the end of the report.

Task 1.A

Using your preferred tool or programming language, implement a software application to solve the *Rare Disease and test problem* explained during the lectures. The program has as input the following probabilities:

P(d) : prior probability of having a disease
P(t | d) : probability that the test is positive given the person has the disease
P(¬t | ¬d) : probability that the test is negative given the person does not have the disease

where the meaning of the variables are:

d: the person has the disease
t: the test is positive

After the values for the previous probabilities are set the program should calculate the rest of necessary probabilities and calculate the probability of having the disease given the test was positive:

P(d | t)

You should be able to easily change the initial probability values and run the program to get the new result.

Task 1.B

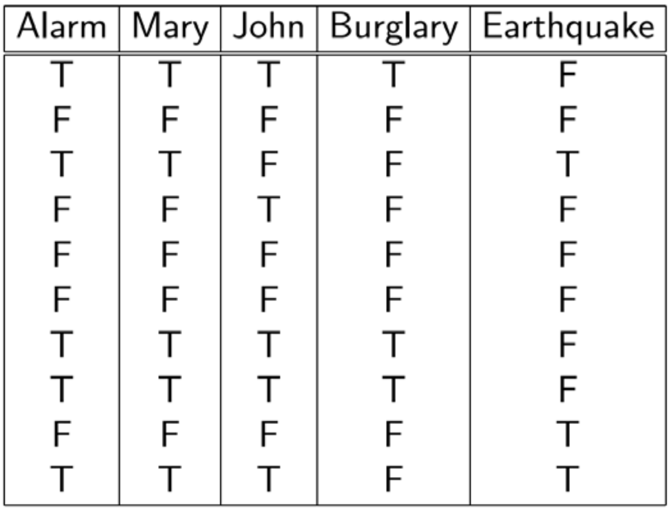
Using your preferred tool or programming language implement a software application to solve the *Burglary problem* explained during the lectures. The program should have as input the dataset shown in the figure below with the following variables:

b: there is a burglary
e: there is an earthquake
a: alarm went off
j: John called
m: Mary called

The program should first learn its parameters from data and then use **inference** to calculate the probability of a burglary given Mary and John called:

P(b | j, m)

Please indicate in your report the methods used by your tool to carry out parameter learning and inference.



Alarm	Mary	John	Burglary	Earthquake
T	T	T	T	F
F	F	F	F	F
T	T	F	F	T
F	F	T	F	F
F	F	F	F	F
F	F	F	F	F
T	T	T	T	F
T	T	T	T	F
F	F	F	F	T
T	T	T	F	T

2. TASK ON HIDDEN MARKOV MODELS

For each one of the following tasks, you should explain in the report the different steps taken to develop the requested software. This will be used by the marker to check inconsistencies or unclear points about the software. The source code must be included in an appendix at the end of the report.

Task

Consider a heater with two possible unknown states, **ON** and **OFF**, placed in a room where the measurable temperature can be Hot, Warm, Cold, or Freezing.
The system is represented by a Hidden Markov Model (HMM) emitting sequences of symbols from the vocabulary **{Warm, Cold, Hot, Freezing}**, and with two hidden states **{ON, OFF}**, which are also equiprobable starting states. The transition probabilities are **0.75** for remaining in a state and **0.25** for switching to the other one.
In state **ON**, the symbols **Warm** and **Cold** are emitted with probability **0.45**, while **Hot** and **Freezing** are emitted with probability **0.05**.
In state **OFF**, the symbols **Warm** and **Cold** are emitted with probability **0.05**, while **Hot** and **Freezing** are emitted with probability **0.45**.

- Implement a software application that, given any input sequence of symbols from the same vocabulary, returns the probability of observing such sequence.

- Describe step-by-step the operations and the numerical results of your software to compute the probability of observing the sequence **Hot-Freezing-Cold-Warm**.