CS5011: Assignment 2 – Uncertainty - Bayesian Networks

Student ID: 220029176

Date of Submission: 15/03/2023

Word Count: 1959

Contents

1. Introduction

The objective of this assignment was to implement, run and evaluate general inference on a general Bayesian network. Three Bayesian networks were provided for the testing and evaluation of the various parts of the practical. An additional four Bayesian networks were created for further testing and evaluation.

The variable elimination algorithm has been implemented for general inference. Various implemented functions can be seen in section 1.1.

The instructions to run the search algorithms can be found in section 1.2. The design and implementation have been documented in section 2. The testing and evaluation of the search algorithms can be found in section 3 and 4 respectively.

1.1 Parts Implemented

Agent	Status
One: Construct a network based on an XML file	Attempted, Tested, Fully Working.
One: Read the query as user input	Attempted, Tested, Fully Working.
One: Calculate the output	Attempted, Tested, Fully Working.
One: Print the output without further information	Attempted, Tested, Fully Working.
Two: Read the order of elimination	Attempted, Tested, Fully Working.
Two: Allow non-binary variables	Attempted, Tested, Fully Working.
Three: Read the evidence set	Attempted, Tested, Fully Working.
Four: Verify a given BN	Attempted, Tested, Fully Working.
Four: Implement Gibbs sampling and Rejection sampling	Not Attempted.

1.2 Compilation and Execution Instructions

1.2.1 Running the Code

To execute the code, please do the following:

1. Download the zip file “CS5011-Assignment2-220029176” and unzip it to a chosen folder location.
2. Open a terminal prompt from within the src folder location.
3. Enter the following command into the terminal “javac A2main.java”
4. Enter the following command into the terminal “java A2main” followed by the desired assessment part and XML file, please see full structure within 1.2.2.

1.2.2 The command structure

The inference system runs on the following structure:

```
java A2main <P1|P2|P3> <BNA.xml|BNB.xml|BNC.xml|BNOne.xml|BNCycle.xml|BNNegative.xml> <Test>
```

<P1|P2|P3>: The part of the practical to test.

<BNA.xml|BNB.xml|BNC.xml>: The name of the XML file to be used.

<Test>: True or False. Whether to show the process of the variable elimination algorithm. (**Optional**)

2. Design and Implementation

2.1 Bayesian Network Model

The Bayesian network model was crafted using information from the provided XML file that includes details on each variable, its associated outcomes, position, and definitions. These definitions comprise of probability tables that relate to a given variable.

Using this file as a guide, a variable class was created to house the variable name, possible outcomes, parents, and probability tables as an instance of the factor class. These variables were then compiled into a Bayesian network class, which maintains a comprehensive list of all the variables involved.

2.2 Factor Implementation

The Factor implementation is an essential component that stores the probability data in a table, enabling variable elimination. The key design elements that make up this implementation include the Sum Out, Assign, Join, Evaluate, and Normalize functions. The Sum Out function eliminates a variable from the current table by summing up all the probabilities for each possible value of the variable. The Assign function eliminates the rows from a given table that do not contain the specified value for the given variable. The Join function iterates through each row of the factors and finds matches for the connecting variables' values. When a match is found, the probabilities of the matching rows are multiplied, and a new row is created. The Evaluate function retrieves the probability value for the provided variable. Finally, the Normalize function sums up all the rows and divides each row by the total, ensuring that the probabilities are normalized and add up to 1.

2.3 Variable Elimination Algorithm

The variable elimination algorithm used in this assignment follows a systematic process that starts with the creation of an elimination order. If the user does not provide an order, the algorithm utilises breadth-first search and all remaining variables are added to the back of the order. The elimination order is then processed one variable at a time, starting with the top of the order. For each variable, all related factors are identified and joined, after which the variable is summed out of the joined factors. The resulting factor replaces the related factors in the system, and this process continues until the elimination order is empty.

Upon reaching this point, the queried variable is normalized and evaluated. In cases where the user provides evidence, the evidence is assigned to the relevant factors before variable elimination begins. This systematic approach ensures that the algorithm can efficiently eliminate all variables in the elimination order and produce accurate results.

2.4 Verifying Bayesian Network

The verification of the Bayesian network involved two distinct steps. Firstly, the network was verified by locating the root nodes and recursively examining each branch originating from them. Any occurrence of an element twice within a given root indicated the presence of a cyclic structure, rendering it illegal. This causes the system to terminate, and an error message would be supplied to the user. Secondly, the verification of the Conditional Probability Tables (CPTs) required ensuring that the probabilities were non-negative and that the sum of probabilities for each variable equalled one. If the probabilities were negative or did not equal one, the system again would terminate, and an error message would be displayed to the user. Verification takes place before a query can be provided and occurs automatically when the system is run. Therefore, no additional parameters are necessary to run this functionality. Furthermore, three new XML files were added to the system to test the functionality and remain within the submitted system, please see 1.2 for instructions on how to run the provided files.

3. Testing

The testing for this project involved manually transcribing each of the factor's functionality. Three new Bayesian networks configurations were developed to test the verification functionality. The tests can be seen below along with the relevant appendices. Additionally, Appendix 23 shows the results from the studres tests provided for this assessment. For further evaluation, the entire system can be evaluated using the **<Test>** parameter as seen in section 1.2.2.

Test Number	Test	Passed?	Proof
1	Reading Bayesian Network	Yes	See Appendix 1 & 2
2	Order of Elimination	Yes	See Appendix 1, 3, 4 & 5
3	Join Functionality	Yes	See Appendix 6 & 7
4	Sum Out Functionality	Yes	See Appendix 8 & 9
5	Assign Functionality	Yes	See Appendix 10 & 11
6	Normalise Functionality	Yes	See Appendix 12 & 13
7	Non-Binary Variable Implementation	Yes	See Appendix 14, 15 & 16
8.1	Verify Bayesian Network – Acyclic	Yes	See Appendix 17 & 18
8.2	Verify Bayesian Network – Non-Negative	Yes	See Appendix 19 & 20
8.3	Verify Bayesian Network – Sum to One	Yes	See Appendix 21 & 22
9	StudRes Tests	Yes	Appendix 23

Test 1 involved basic reading and parsing the XML file to the Bayesian Network and Variable classes. The provided appendices showcase a working transition of the BNA.xml file to the correct classes, laying the foundation for the rest of the system. Test 2 showed the automation variable ordering for the elimination. From Appendix 5, we can see that the automatic ordering for BNC is P, R, Q, S, V, U. This order is based on the initial query of Z = true, this ordering is efficient and close to the optimal solution as it is based on breadth-first search. However, based on the diagram of BNC as seen in Appendix 4, the variable U is redundant to the final solution. Therefore, wasting computational time to reach a solution. The inclusion of a redundancy checker should be incorporated into future versions of the system.

Tests 3 to 6 assess the functionality of the system. All functionality was tested on paper, using techniques established during lectures. Testing by hand enhances the understanding of key processes within each function as well as enabling critical analysis of the results. All the functions tested were compared with the handwritten copy for accuracy. From this evaluation, all the functions work according to the lecture material and the transcribed analysis.

Test 7 evaluates the non-binary requirement of the system. This requirement was built into the foundation of the system and was always in mind throughout the development of the Factor functionality. A new XML file was built to test this requirement called BND.xml. This Bayesian network incorporates a variable A with 3 outcomes: Sunny, Rainy and Cloudy. This network was tested against each step within the variable elimination algorithm with evidence to guarantee the effectiveness of the system for non-binary networks. The system passed this test with the non-binary system and the whole process can be seen in Appendix 15.

Test 8 was broken into three parts to extensively test the extension functionality. The verification of a Bayesian network required testing for an acyclic system as well as, testing that all probabilities sum to one, and are non-negative. Each test required the development of a new Bayesian network as all the provided Bayesian networks are legal. For the acyclic test, BNCycle.xml was created with a cycle between variable B and variable D. This continuous cycle between B, C and D is illegal within a Bayesian network. For the non-negative test, BNNegative.xml was created where negative values were added to variable B. For the sum to one test, BNOne.xml was created with a sum value of 1.1 appearing in variable B. When running these Bayesian networks, an error occurs, and a message is sent to the user's terminal, whilst the system terminates. This is the intended functionality, showing this verification system passes the necessary checks. However, these tests are limited to only basic instances of a Bayesian network. For a full evaluation of the verification process, a more extensive Bayesian network will need to be developed.

Test 9 utilised the provided studres tests, these tests provided a limited view of the system whilst testing essential components and functionalities. From Appendix 23, it is clear that the system developed suited the required functionality. Therefore, matching the initial specification and fulfilling the assessment criteria.

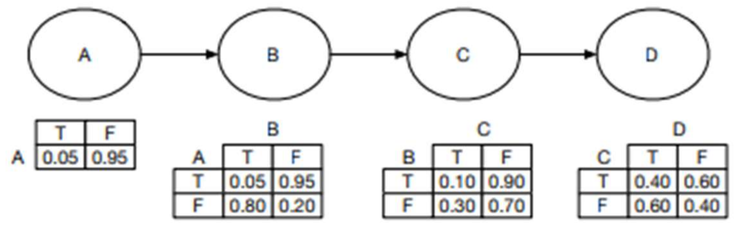
4. Evaluation

In this assessment, a functional and effective system was developed and tested for conducting Bayesian network analysis. The initial tests established a solid foundation for the system, including a successful transition of the XML file and an efficient automation variable ordering for the elimination process. Subsequent tests evaluated the functionality of the system in various scenarios, including a non-binary system and extensive verification of Bayesian networks. The system passed all tests and proved to be an effective tool for conducting Bayesian network analysis. However, this report also highlights areas for future improvement, including the incorporation of a redundancy checker and the need for more extensive testing of the verification process.

The system developed as part of this assessment provides exact inference using the variable elimination algorithm and works with both binary and non-binary variables. The comments within the code and design provide a detailed analysis of each functionality, demonstrating a deep understanding of Bayesian networks. Initially, a mechanical approach was taken to develop the system, but upon reflection and reviewing the lecture materials, a more algorithmic approach was adopted. While the factor utilised for this assessment was based on a list of lists, an approach advocated in the lecture material relating to a hashmap system may have improved code appearance, and space and time complexity. A future development of the system would include this approach, and an evaluation of efficiency and performance would be conducted.

Another extension to this system would include Gibbs sampling and Rejection sampling, which were requested from the assessment but were not fulfilled due to time constraints. The comparison between exact inference and approximate inference would have provided further understanding of the system's performance. Despite the areas for improvement, the system developed met the initial specifications and fulfilled the assessment criteria, making it a valuable contribution to the field of Bayesian network analysis.

5. Appendices



	T	F
A	0.05	0.95

B

A	T	F
T	0.05	0.95
F	0.80	0.20

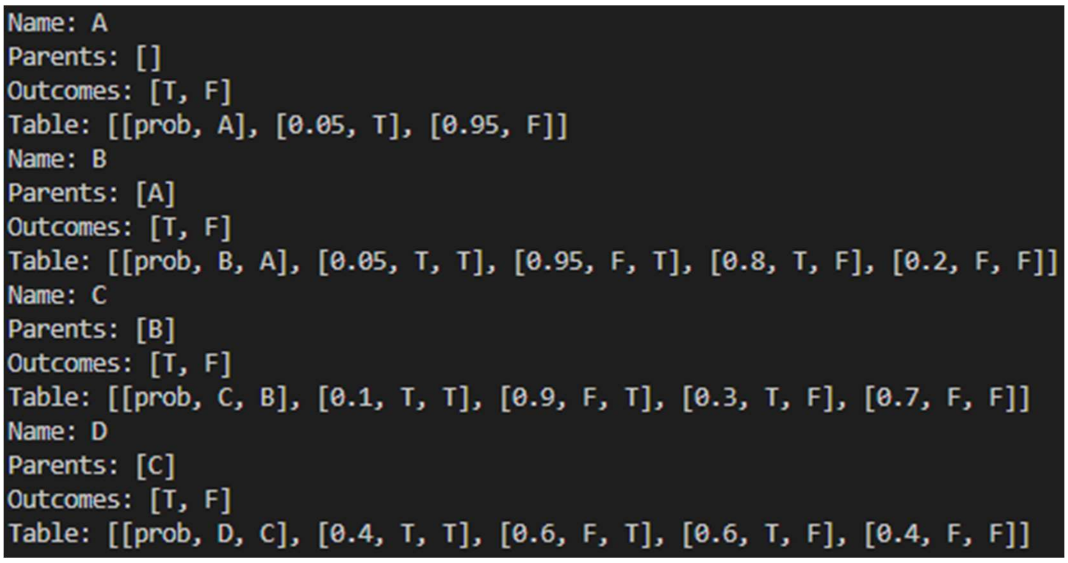
C

B	T	F
T	0.10	0.90
F	0.30	0.70

D

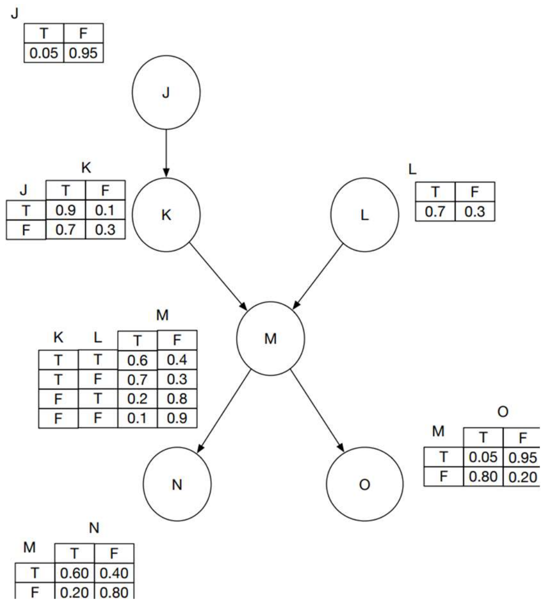
C	T	F
T	0.40	0.60
F	0.60	0.40

Appendix 1: BNA Diagram



```
Name: A
Parents: []
Outcomes: [T, F]
Table: [[prob, A], [0.05, T], [0.95, F]]
Name: B
Parents: [A]
Outcomes: [T, F]
Table: [[prob, B, A], [0.05, T, T], [0.95, F, T], [0.8, T, F], [0.2, F, F]]
Name: C
Parents: [B]
Outcomes: [T, F]
Table: [[prob, C, B], [0.1, T, T], [0.9, F, T], [0.3, T, F], [0.7, F, F]]
Name: D
Parents: [C]
Outcomes: [T, F]
Table: [[prob, D, C], [0.4, T, T], [0.6, F, T], [0.6, T, F], [0.4, F, F]]
```

Appendix 2: Bayesian Network instance of BNA



J

T	F
0.05	0.95

K

J	T	F
T	0.9	0.1
F	0.7	0.3

L

T	F
0.7	0.3

M

K	L	T	F
T	T	0.6	0.4
T	F	0.7	0.3
F	T	0.2	0.8
F	F	0.1	0.9

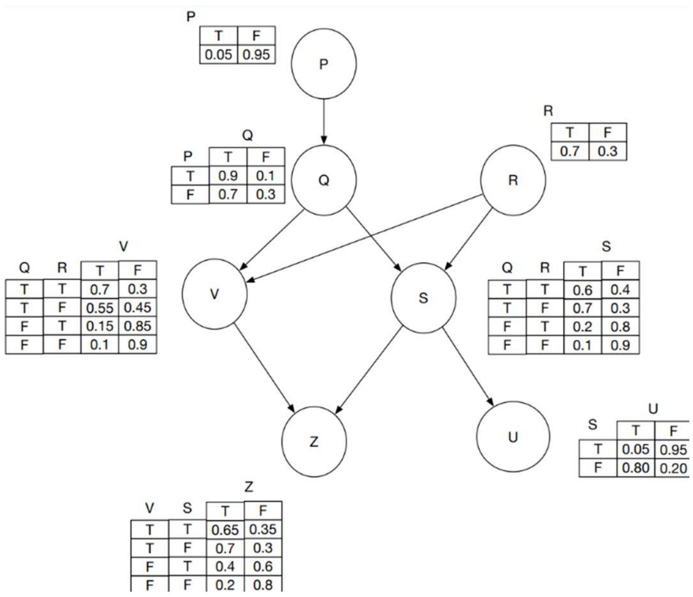
O

M	T	F
T	0.05	0.95
F	0.80	0.20

N

M	T	F
T	0.60	0.40
F	0.20	0.80

Appendix 3: BNB Diagram



P

T	F
0.05	0.95

R

T	F
0.7	0.3

Q

P	T	F
T	0.9	0.1
F	0.7	0.3

V

Q	R	T	F
T	T	0.7	0.3
T	F	0.55	0.45
F	T	0.15	0.85
F	F	0.1	0.9

S

Q	R	T	F
T	T	0.6	0.4
T	F	0.7	0.3
F	T	0.2	0.8
F	F	0.1	0.9

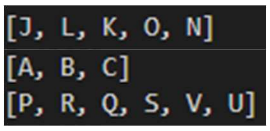
U

S	T	F
T	0.05	0.95
F	0.80	0.20

Z

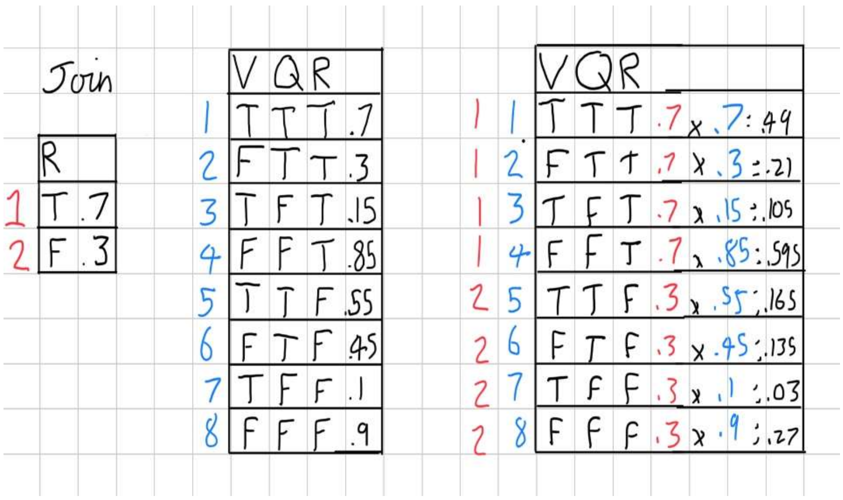
V	S	T	F
T	T	0.65	0.35
T	F	0.7	0.3
F	T	0.4	0.6
F	F	0.2	0.8

Appendix 4: BNC Diagram



```
[J, L, K, O, N]
[A, B, C]
[P, R, Q, S, V, U]
```

Appendix 5: Implicit Order of Elimination for BNB, where Query=M, BNA, where Query=D, and BNC, where Query=Z respectively



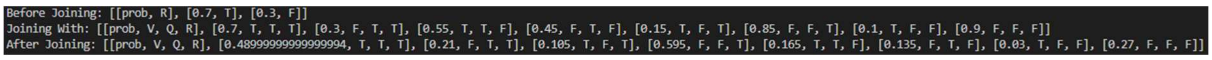
Join

	R	
1	T	.7
2	F	.3

	V	Q	R	
1	T	T	T	.7
2	F	T	T	.3
3	T	F	T	.15
4	F	F	T	.85
5	T	T	F	.55
6	F	T	F	.45
7	T	F	F	.1
8	F	F	F	.9

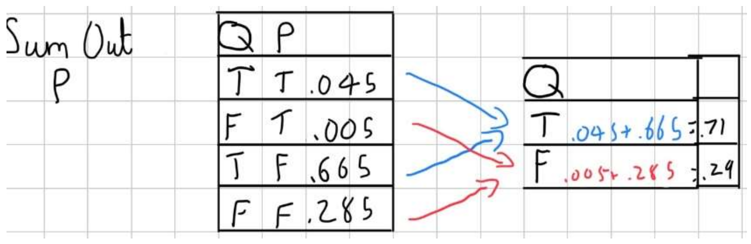
		V	Q	R	
1	1	T	T	T	.7 x .7 : .49
1	2	F	T	T	.7 x .3 : .21
1	3	T	F	T	.7 x .15 : .105
1	4	F	F	T	.7 x .85 : .595
2	5	T	T	F	.3 x .55 : .165
2	6	F	T	F	.3 x .45 : .135
2	7	T	F	F	.3 x .1 : .03
2	8	F	F	F	.3 x .9 : .27

Appendix 6: Handwritten Example of the Join Functionality. Joining the V,Q, R Factor with the R Factor Using BNC



```
Before Joining: [[prob, R], [0.7, T], [0.3, F]]
Joining With: [[prob, V, Q, R], [0.7, T, T, T], [0.3, F, T, T], [0.55, T, T, F], [0.45, F, T, F], [0.15, T, F, T], [0.85, F, F, T], [0.1, T, F, F], [0.9, F, F, F]]
After Joining: [[prob, V, Q, R], [0.48999999999999994, T, T, T], [0.21, F, T, T], [0.105, T, F, T], [0.595, F, F, T], [0.165, T, T, F], [0.135, F, T, F], [0.03, T, F, F], [0.27, F, F, F]]
```

Appendix 7: Joining Functionality between R Factor and V, Q, R Factor

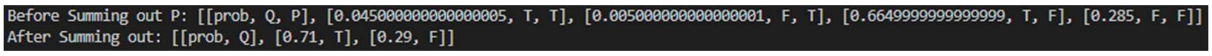


Sum Out
P

Q	P	
T	T	.045
F	T	.005
T	F	.665
F	F	.285

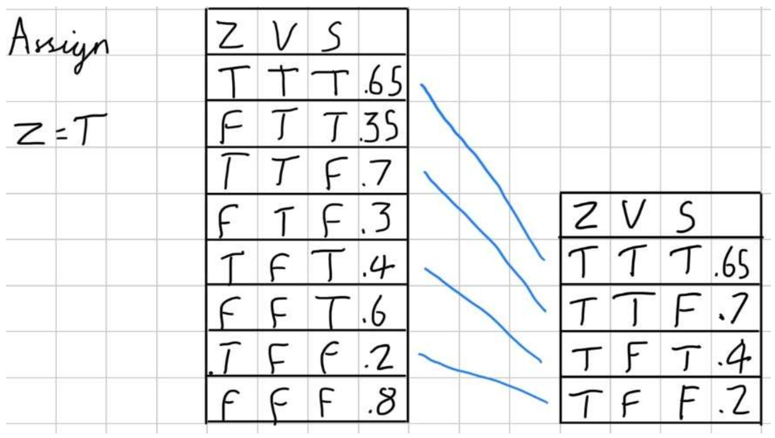
Q		
T	.045+.665=	.71
F	.005+.285=	.29

Appendix 8: Handwritten Example of the Sum Out Functionality. Example Shows the Summing Out of Factor P for the Q Factor



```
Before Summing out P: [[prob, Q, P], [0.045000000000000005, T, T], [0.005000000000000001, F, T], [0.6649999999999999, T, F], [0.285, F, F]]
After Summing out: [[prob, Q], [0.71, T], [0.29, F]]
```

Appendix 9: Sum Out Functionality of P on the Q Factor



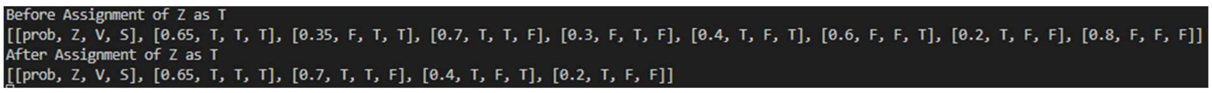
Assign

Z=T

Z	V	S	
T	T	T	.65
F	T	T	.35
T	T	F	.7
F	T	F	.3
T	F	T	.4
F	F	T	.6
T	F	F	.2
F	F	F	.8

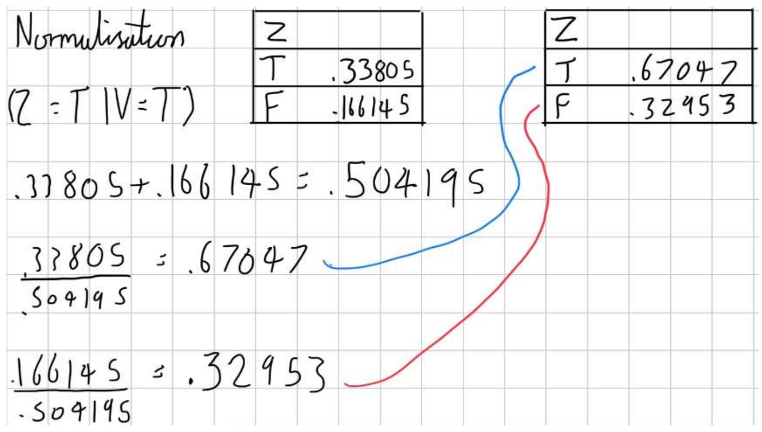
Z	V	S	
T	T	T	.65
T	T	F	.7
T	F	T	.4
T	F	F	.2

Appendix 10: Handwritten Example of the Assign Functionality. Example Shows the Assigning of Factor Z = True for the Z Factor



```
Before Assignment of Z as T
[[prob, Z, V, S], [0.65, T, T, T], [0.35, F, T, T], [0.7, T, T, F], [0.3, F, T, F], [0.4, T, F, T], [0.6, F, F, T], [0.2, T, F, F], [0.8, F, F, F]]
After Assignment of Z as T
[[prob, Z, V, S], [0.65, T, T, T], [0.7, T, T, F], [0.4, T, F, T], [0.2, T, F, F]]
```

Appendix 11: Assignment Functionality of Z=T on the Z Factor



Normalisation

(Z = T | V = T)

Z	
T	.33805
F	.166145

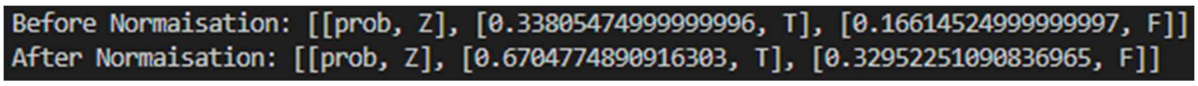
Z	
T	.67047
F	.32953

$.33805 + .166145 = .504195$

$\frac{.33805}{.504195} = .67047$

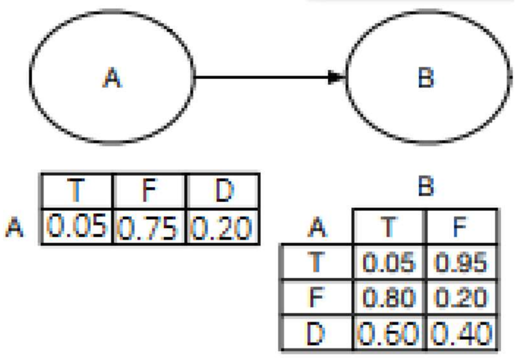
$\frac{.166145}{.504195} = .32953$

Appendix 12: Handwritten Example of the Normalisation Functionality. The Example Shows the Normalisation of the Z Table When Factor V = True



```
Before Normaisation: [[prob, Z], [0.33805474999999996, T], [0.16614524999999997, F]]
After Normaisation: [[prob, Z], [0.6704774890916303, T], [0.32952251090836965, F]]
```

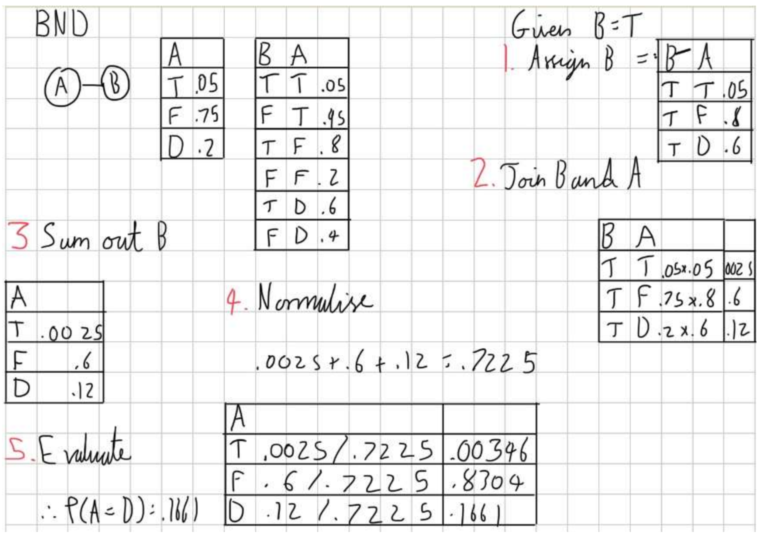
Appendix 13: Normalisation Functionality for Factor Z



	T	F	D
A	0.05	0.75	0.20

	B	
A	T	F
T	0.05	0.95
F	0.80	0.20
D	0.60	0.40

Appendix 14: BND Diagram



A	
T	.05
F	.75
D	.2

B	A	
T	T	.05
F	T	.95
T	F	.8
F	F	.2
T	D	.6
F	D	.4

Given B=T

1. Assign B =

~~B~~	A	
T	T	.05
T	F	.8
T	D	.6

2. Join B and A

B	A		
T	T	.05x.05	.0025
T	F	.75x.8	.6
T	D	.2x.6	.12

3 Sum out B

A	
T	.0025
F	.6
D	.12

4. Normalise

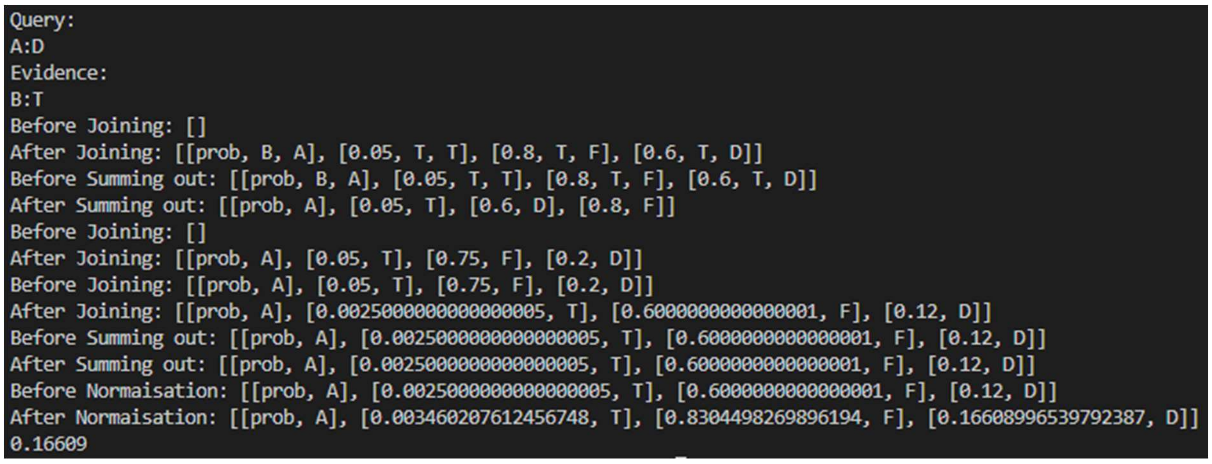
.0025+.6+.12 = .7225

5. Evaluate

A		
T	.0025/.7225	.00346
F	.6/.7225	.8304
D	.12/.7225	.1661

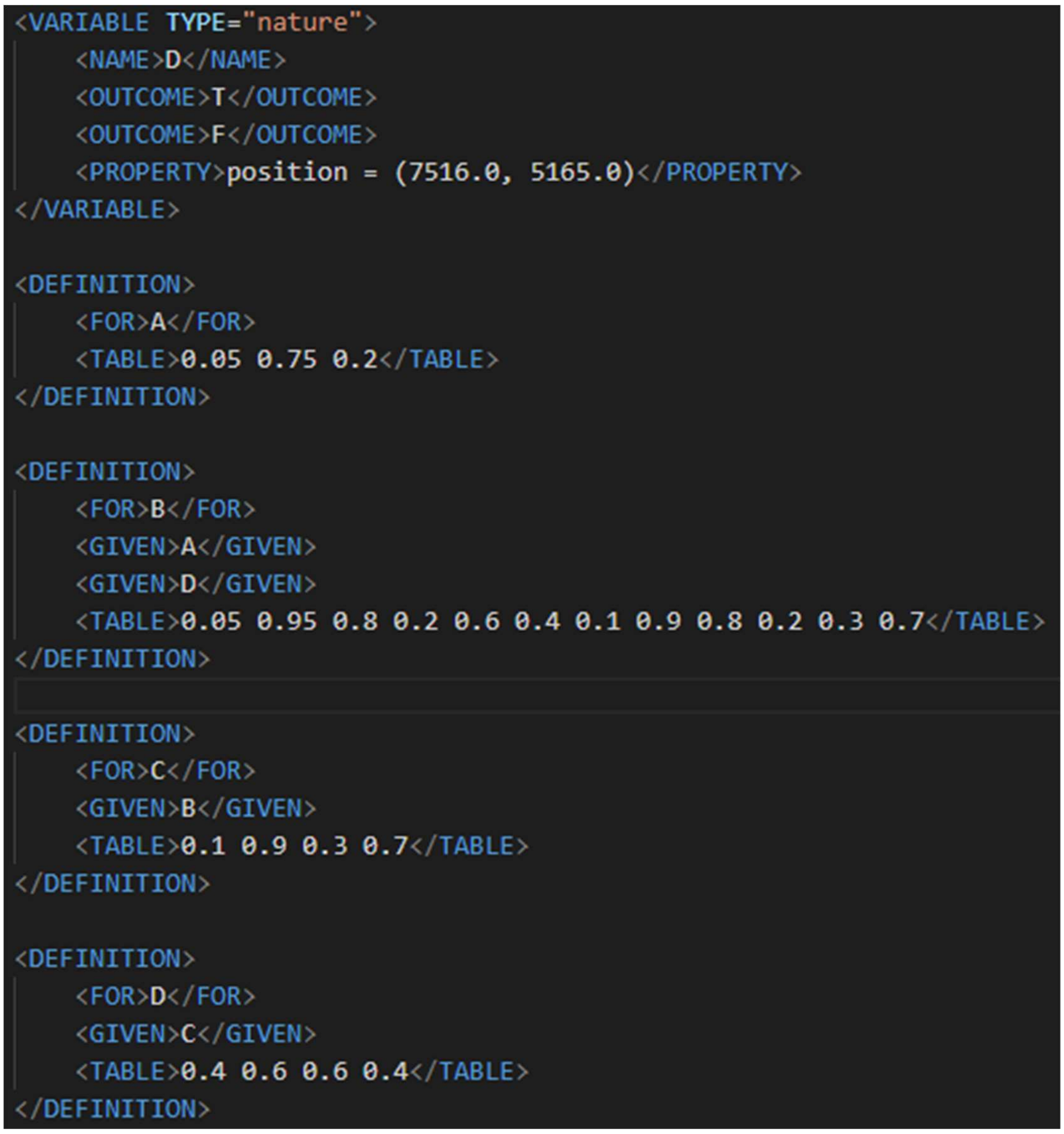
$\therefore P(A=D) = .1661$

Appendix 15: Handwritten Example of the Full Variable Elimination Algorithm for the Non-Binary System, BND



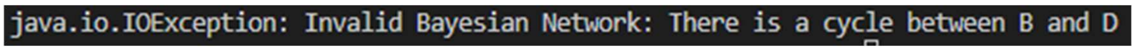
```
Query:
A:D
Evidence:
B:T
Before Joining: []
After Joining: [[prob, B, A], [0.05, T, T], [0.8, T, F], [0.6, T, D]]
Before Summing out: [[prob, B, A], [0.05, T, T], [0.8, T, F], [0.6, T, D]]
After Summing out: [[prob, A], [0.05, T], [0.6, D], [0.8, F]]
Before Joining: []
After Joining: [[prob, A], [0.05, T], [0.75, F], [0.2, D]]
Before Joining: [[prob, A], [0.05, T], [0.75, F], [0.2, D]]
After Joining: [[prob, A], [0.0025000000000000005, T], [0.6000000000000001, F], [0.12, D]]
Before Summing out: [[prob, A], [0.0025000000000000005, T], [0.6000000000000001, F], [0.12, D]]
After Summing out: [[prob, A], [0.0025000000000000005, T], [0.6000000000000001, F], [0.12, D]]
Before Normaisation: [[prob, A], [0.0025000000000000005, T], [0.6000000000000001, F], [0.12, D]]
After Normaisation: [[prob, A], [0.003460207612456748, T], [0.8304498269896194, F], [0.16608996539792387, D]]
0.16609
```

Appendix 16: Full Working Example of the Non-Binary System, BND



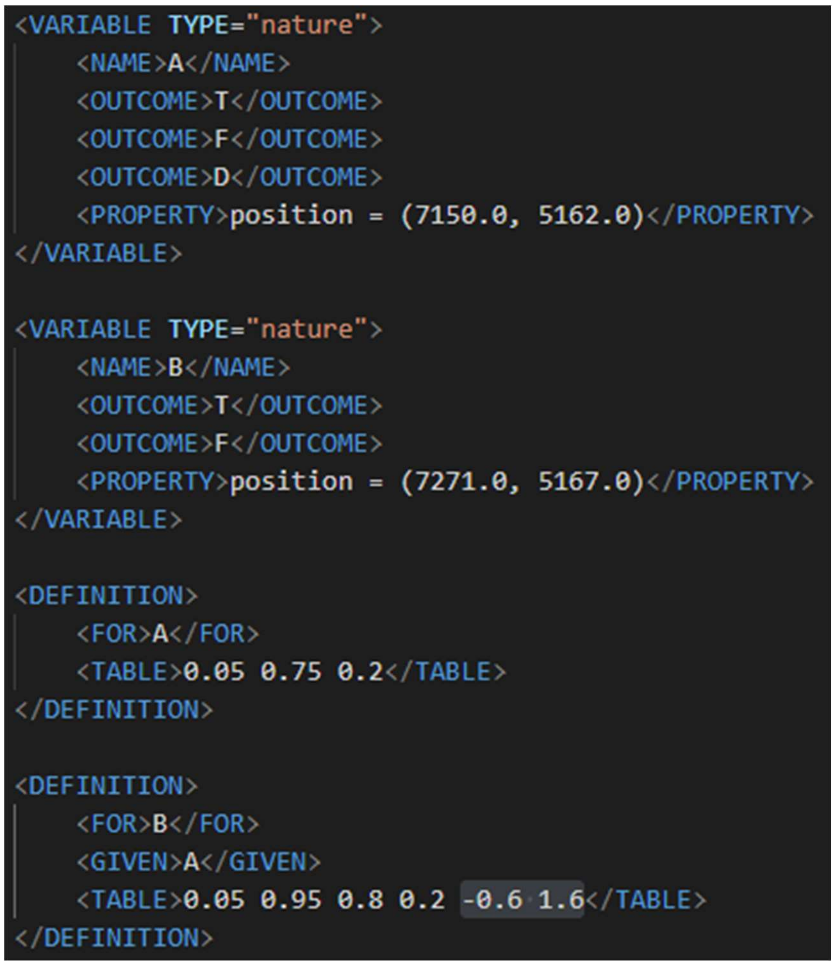
```xml
<VARIABLE TYPE="nature">
    <NAME>D</NAME>
    <OUTCOME>T</OUTCOME>
    <OUTCOME>F</OUTCOME>
    <PROPERTY>position = (7516.0, 5165.0)</PROPERTY>
</VARIABLE>

<DEFINITION>
    <FOR>A</FOR>
    <TABLE>0.05 0.75 0.2</TABLE>
</DEFINITION>

<DEFINITION>
    <FOR>B</FOR>
    <GIVEN>A</GIVEN>
    <GIVEN>D</GIVEN>
    <TABLE>0.05 0.95 0.8 0.2 0.6 0.4 0.1 0.9 0.8 0.2 0.3 0.7</TABLE>
</DEFINITION>

<DEFINITION>
    <FOR>C</FOR>
    <GIVEN>B</GIVEN>
    <TABLE>0.1 0.9 0.3 0.7</TABLE>
</DEFINITION>

<DEFINITION>
    <FOR>D</FOR>
    <GIVEN>C</GIVEN>
    <TABLE>0.4 0.6 0.6 0.4</TABLE>
</DEFINITION>
```

Appendix 17: The XML File for the BNCycle, where Variable B and Variable D form a cycle.



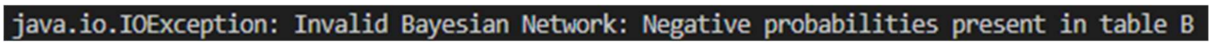
```
java.io.IOException: Invalid Bayesian Network: There is a cycle between B and D
```

Appendix 18: The Error Message Generated When BNCycle is Parsed Through the Verification Method



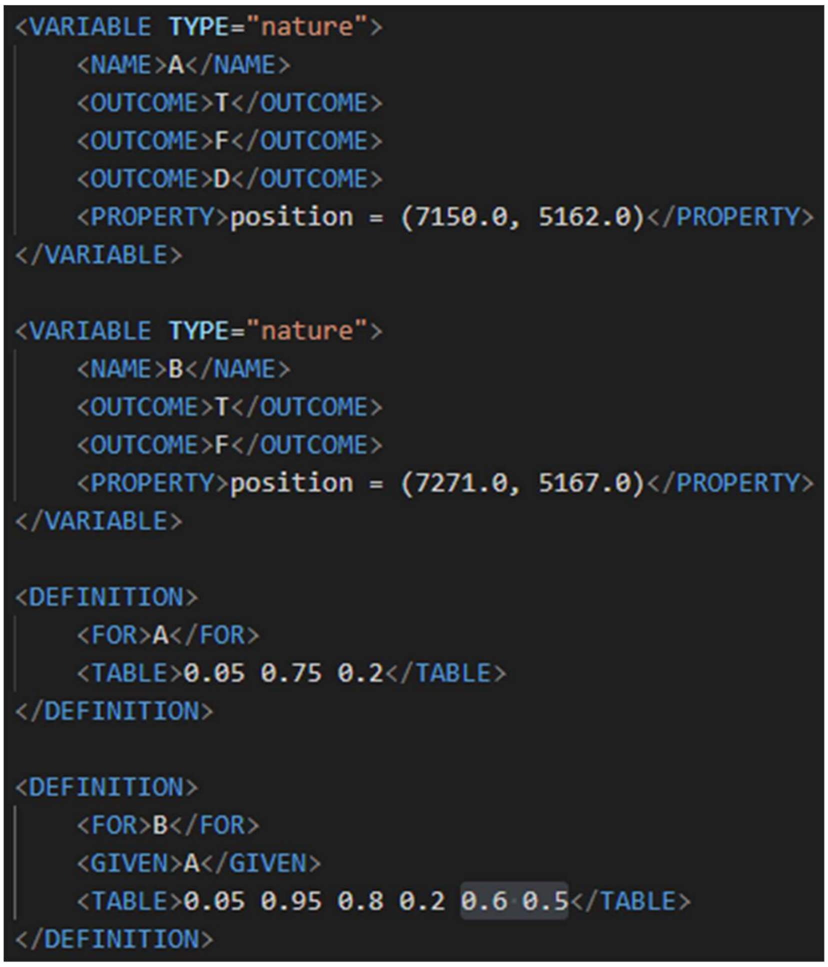
```xml
<VARIABLE TYPE="nature">
    <NAME>A</NAME>
    <OUTCOME>T</OUTCOME>
    <OUTCOME>F</OUTCOME>
    <OUTCOME>D</OUTCOME>
    <PROPERTY>position = (7150.0, 5162.0)</PROPERTY>
</VARIABLE>

<VARIABLE TYPE="nature">
    <NAME>B</NAME>
    <OUTCOME>T</OUTCOME>
    <OUTCOME>F</OUTCOME>
    <PROPERTY>position = (7271.0, 5167.0)</PROPERTY>
</VARIABLE>

<DEFINITION>
    <FOR>A</FOR>
    <TABLE>0.05 0.75 0.2</TABLE>
</DEFINITION>

<DEFINITION>
    <FOR>B</FOR>
    <GIVEN>A</GIVEN>
    <TABLE>0.05 0.95 0.8 0.2 -0.6 1.6</TABLE>
</DEFINITION>
```

Appendix 19: The XML File for the BNNegative, where CPT B contains a negative value



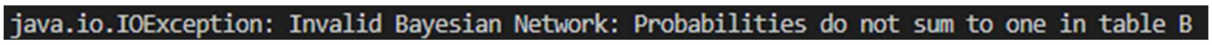
```
java.io.IOException: Invalid Bayesian Network: Negative probabilities present in table B
```

Appendix 20: The error message generated when BNNegative is parsed through the verification method



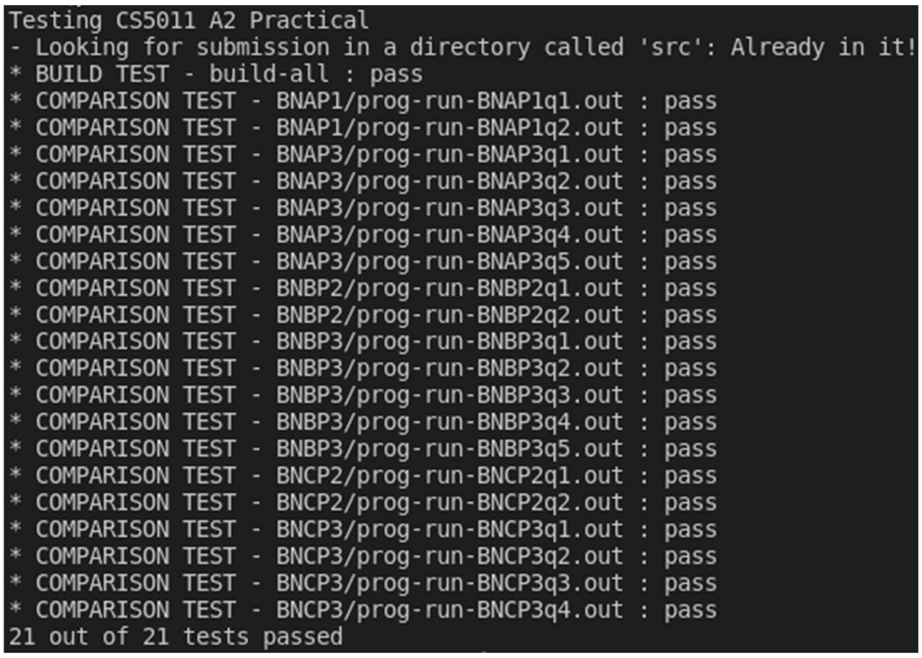
```xml
<VARIABLE TYPE="nature">
    <NAME>A</NAME>
    <OUTCOME>T</OUTCOME>
    <OUTCOME>F</OUTCOME>
    <OUTCOME>D</OUTCOME>
    <PROPERTY>position = (7150.0, 5162.0)</PROPERTY>
</VARIABLE>

<VARIABLE TYPE="nature">
    <NAME>B</NAME>
    <OUTCOME>T</OUTCOME>
    <OUTCOME>F</OUTCOME>
    <PROPERTY>position = (7271.0, 5167.0)</PROPERTY>
</VARIABLE>

<DEFINITION>
    <FOR>A</FOR>
    <TABLE>0.05 0.75 0.2</TABLE>
</DEFINITION>

<DEFINITION>
    <FOR>B</FOR>
    <GIVEN>A</GIVEN>
    <TABLE>0.05 0.95 0.8 0.2 0.6 0.5</TABLE>
</DEFINITION>
```

Appendix 21: The XML File for the BNOne, where CPT B has a summed probability value of 1.1



```
java.io.IOException: Invalid Bayesian Network: Probabilities do not sum to one in table B
```

Appendix 22: The Error message generated when BNOne is parsed through the verification method



```
Testing CS5011 A2 Practical
- Looking for submission in a directory called 'src': Already in it!
* BUILD TEST - build-all : pass
* COMPARISON TEST - BNAP1/prog-run-BNAP1q1.out : pass
* COMPARISON TEST - BNAP1/prog-run-BNAP1q2.out : pass
* COMPARISON TEST - BNAP3/prog-run-BNAP3q1.out : pass
* COMPARISON TEST - BNAP3/prog-run-BNAP3q2.out : pass
* COMPARISON TEST - BNAP3/prog-run-BNAP3q3.out : pass
* COMPARISON TEST - BNAP3/prog-run-BNAP3q4.out : pass
* COMPARISON TEST - BNAP3/prog-run-BNAP3q5.out : pass
* COMPARISON TEST - BNBP2/prog-run-BNBP2q1.out : pass
* COMPARISON TEST - BNBP2/prog-run-BNBP2q2.out : pass
* COMPARISON TEST - BNBP3/prog-run-BNBP3q1.out : pass
* COMPARISON TEST - BNBP3/prog-run-BNBP3q2.out : pass
* COMPARISON TEST - BNBP3/prog-run-BNBP3q3.out : pass
* COMPARISON TEST - BNBP3/prog-run-BNBP3q4.out : pass
* COMPARISON TEST - BNBP3/prog-run-BNBP3q5.out : pass
* COMPARISON TEST - BNCP2/prog-run-BNCP2q1.out : pass
* COMPARISON TEST - BNCP2/prog-run-BNCP2q2.out : pass
* COMPARISON TEST - BNCP3/prog-run-BNCP3q1.out : pass
* COMPARISON TEST - BNCP3/prog-run-BNCP3q2.out : pass
* COMPARISON TEST - BNCP3/prog-run-BNCP3q3.out : pass
* COMPARISON TEST - BNCP3/prog-run-BNCP3q4.out : pass
21 out of 21 tests passed
```

Appendix 23: The summary of the StudRes Test Results