SCHOOL OF COMPUTING (SOC)

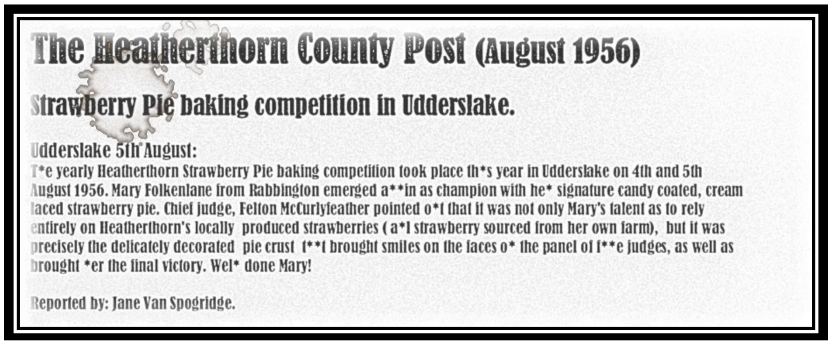
Diploma in Applied AI and Analytics

ST1507 DATA STRUCTURES AND ALGORITHMS (AI)

2025/26 SEMESTER 1
ASSIGNMENT TWO (CA2)

~ Restoring old newspapers ~

(using prefix tries & predictive text analysis)



The Heatherthorn County Post (August 1956)

Strawberry Pie baking competition in Udderslake.

Udderslake 5th August:
T*e yearly Heatherthorn Strawberry Pie baking competition took place th*s year in Udderslake on 4th and 5th August 1956. Mary Folkenlane from Rabbington emerged a**in as champion with he* signature candy coated, cream laced strawberry pie. Chief judge, Felton McCurlyfeather pointed o*t that it was not only Mary's talent as to rely entirely on Heatherthorn's locally produced strawberries (a*l strawberry sourced from her own farm), but it was precisely the delicately decorated pie crust t**t brought smiles on the faces o* the panel of f**e judges, as well as brought *er the final victory. Wel* done Mary!

Reported by: Jane Van Spogridge.

Objective of Assignment

For this assignment you will have an opportunity to apply all that you have learnt with regards to data structures, algorithms, and object-oriented programming as to develop an application that can help the town council of Heatherthorn County to restore past newspaper editions from the *Heatherthorn County Post.*

Background

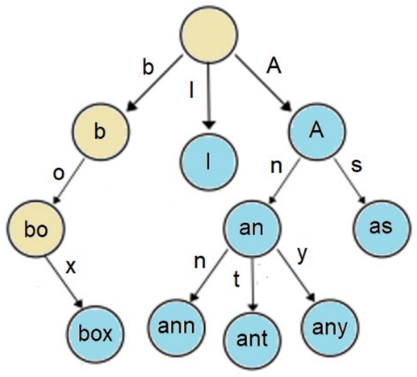
The town council of Heatherthorn County is in possession of a large archive of past editions from the *Heatherthorn County Post* stemming from the fifties and sixties. They are in the process of digitizing the old newspapers, that have deteriorated over the last decades due to humidity, and the less than perfect conditions that they were stored. The newspapers are scanned using OCR technology (Optical Character Recognition) and converted into text files. During the OCR process, all those characters that were not identifiable have been labeled with an asterix character ('*').

Your job as team

As a team of AI programmers, you have been roped in to implement an application that can help to restore the old editions of the Heatherthorn Post. Your application will be making use of *prefix tries* that can then be used to restore words that have missing characters.

What is a prefix trie?

A *Prefix trie* (‘trie’ is pronounced as 'tray') is a special type of tree that can be used to store words that share common prefixes (such as for instance ‘Car’,’ Cat’, ‘Catapult’,’ Catamaran’, words that all share a common prefix ‘Ca’). When compared to ordinary binary search trees (BST), prefix tries provide better support for implementing string-searching algorithms.



Above is an example of a trie that stores 8 words (referred to as ‘keys’). In this case the keys stored are ‘A’, ‘box’, ’an’, ’as’, ’I’, ’any’, ’ant’ and ’ann’. The blue nodes are referred to as ‘terminal nodes’ as they are associated with complete English words (keywords). The brown nodes represent merely prefixes of English words. Optionally, the terminal nodes can be used to store a value for a keyword, for instance the frequency of a keyword.

Prefix tries provide better support for implementing string-searching algorithms specifically those involving predictive text analyses, approximate string matching, and spelling checking.

For this assignment the objective is to develop an application that allows a user to construct, prefix tries, and then to utilize them for text prediction as to restore historical newspapers.

Instructions and Guidelines:

1. This is a group assignment (you will work in pairs, only one group of 3 would be allowed if there is an odd number of total students).

2. This assignment accounts for **40%** of your final grade.

3. The assignment comprises a group component (70%) and an individual component (30%).

4. The submission date is **Wednesday 13 August 1:00 pm**.

5. The development will be carried out in Python using Anaconda.

6. The demonstrations/interviews will be conducted during the DSAA lessons in week 17/18. You are expected to explain your code and program logic. Take note that the interview is compulsory.

7. **50% of marks** will be deducted for submission of assignment within **ONE** calendar day after the deadline. **No marks shall** be awarded for assignments submitted **more than one day** after the deadline.

Warning: Plagiarism means passing off as one's own the ideas, works, writings, etc., which belong to another person. In accordance with this definition, you are committing plagiarism if you copy the work of another person and turning it in as your own, even if you would have the permission of that person.
Plagiarism is a serious offence, and if you are found to have committed, aided, and/or abetted the offence of plagiarism, disciplinary action will be taken against you. If you are guilty of plagiarism, you may fail all modules in the semester or even be liable for expulsion.

<u>Overview of the basic features of the system</u>

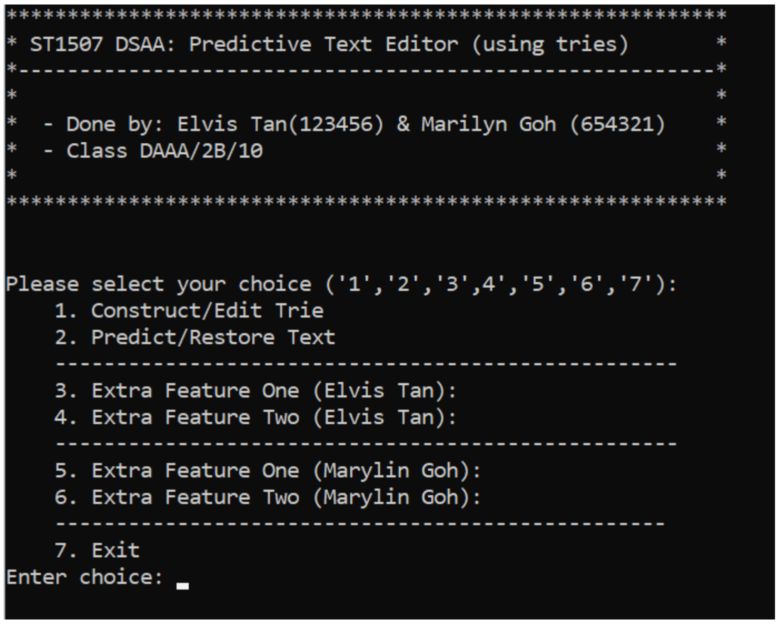
Your group is tasked to implement an application that allows a user to construct, read in, and edit tries, and then to utilize these tries for text prediction.

Your application should support the following features:

- The user should be able add, delete and find keywords in a prefix trie, as well as display a prefix trie.
- The user should be able to write a prefix trie to file, as well as construct a prefix trie from keywords as well as write the keywords from a prefix trie back to file.
- The user should be able to list all possible matching keywords for a keyword with wild characters, as well as restore a keyword with wild characters with best match.
- The user should be able restore all keywords with wild characters from a text file and restore them using all matching keywords, as well as restore them using best keyword matches.

<u>Selection menu</u>

When the application starts, the user will be presented with a menu as is shown below, allowing the user to choose from 7 options (1','2','3','4','5','7').



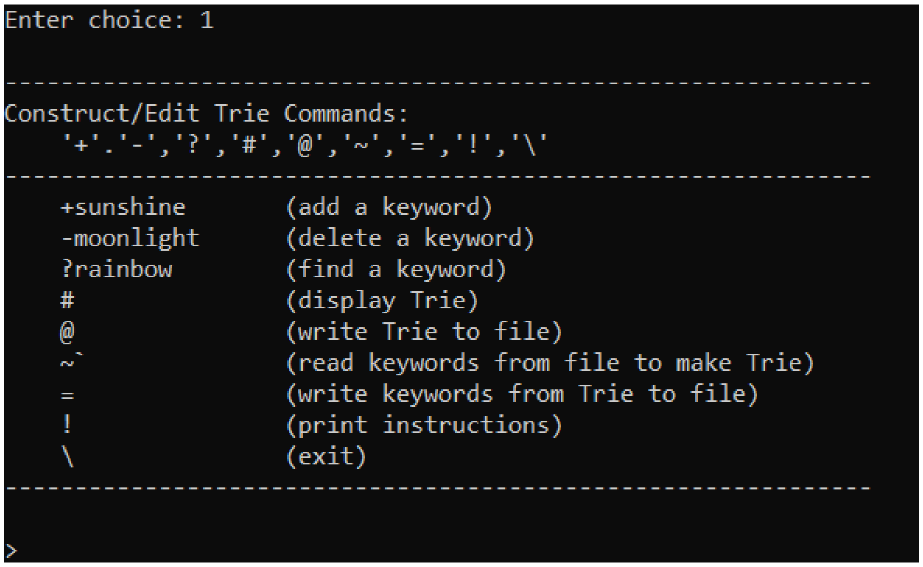
```
*********************************************************
* ST1507 DSAA: Predictive Text Editor (using tries)     *
*-------------------------------------------------------*
*                                                       *
*  - Done by: Elvis Tan(123456) & Marilyn Goh (654321)  *
*  - Class DAAA/2B/10                                   *
*                                                       *
*********************************************************


Please select your choice ('1','2','3',4','5','6','7'):
    1. Construct/Edit Trie
    2. Predict/Restore Text
    -------------------------------------------------
    3. Extra Feature One (Elvis Tan):
    4. Extra Feature Two (Elvis Tan):
    -------------------------------------------------
    5. Extra Feature One (Marylin Goh):
    6. Extra Feature Two (Marylin Goh):
    ------------------------------------------------
    7. Exit
Enter choice: _
```

- Take note you must follow the above format, please ensure you display the names and IDs of all group members, as well as the correct class.
- The user will be able to repeatedly select options from the menu, until he/she selects option 7 after which the application will terminate.

(*) Take note you need to replace 'Elvis Tan' and 'Marilyn Goh' with your own names.

Constructing & Editing Tries

- By selecting option One, the user will enter a command prompt panel that can be used for constructing and editing tries.



```
Enter choice: 1

--------------------------------------------------------------
Construct/Edit Trie Commands:
    '+'.'-','?','#','@','~','=','!','\'
--------------------------------------------------------------
    +sunshine       (add a keyword)
    -moonlight      (delete a keyword)
    ?rainbow        (find a keyword)
    #               (display Trie)
    @               (write Trie to file)
    ~`              (read keywords from file to make Trie)
    =               (write keywords from Trie to file)
    !               (print instructions)
    \               (exit)
--------------------------------------------------------------


>
```

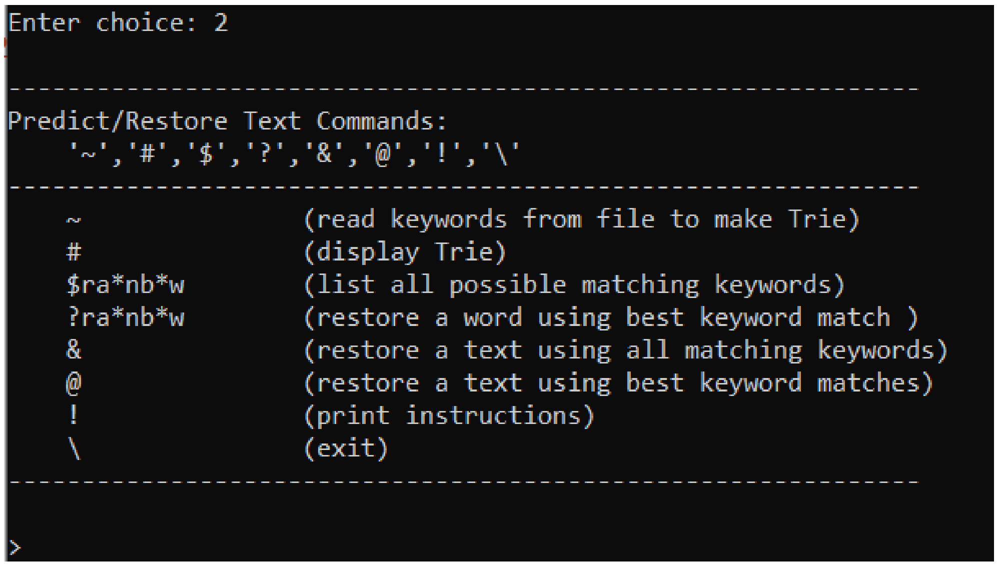
The commands and respective actions available in this command prompt are listed in below table:

Construct/Edit Trie Commands	**Action**
+	*Adds a new keyword to the current Trie.*
-	*Deletes a keyword from the current Trie.*
?	*Searches for a keyword in the current Trie.*
#	*Displays the current Trie on the screen.*
@	*Writes the current Trie to a file.*
~	*Reads keywords from a file to make a new Trie (thereby clearing the current Trie).*
=	*Writes all the keywords from the current Trie to a file.*
!	*Prints the instructions for the various Construct/Edit Trie Commands.*
\	*Exits the Edit Trie Command Prompt and returns to the main menu.*

Appendix A gives an example of a complete session whereby we construct a Trie from scratch using the various commands.

Predicting & Restoring Text

- By selecting option Two, the user will enter a command prompt panel that can be used for predicting and restoring text based on the currently loaded prefix trie.



```
Enter choice: 2

------------------------------------------------------------
Predict/Restore Text Commands:
    '~','#','$','?','&','@','!','\'
------------------------------------------------------------
    ~               (read keywords from file to make Trie)
    #               (display Trie)
    $ra*nb*w        (list all possible matching keywords)
    ?ra*nb*w        (restore a word using best keyword match )
    &               (restore a text using all matching keywords)
    @               (restore a text using best keyword matches)
    !               (print instructions)
    \               (exit)
------------------------------------------------------------

>
```

The commands and respective actions available in this command prompt are listed in the table below:

Predict/Restore Text Commands	Action
~	*Reads keywords from a file as to make a new prefix trie (take note, thereby it clears the current prefix trie).*
#	*Displays the current prefix trie on the screen.*
$	*Lists all possible matching keywords.*
?	*Restores a word using the best keyword match as based on word frequencies.*
&	*Restores a text using all matching keywords.*
@	*Restores a text using the best keyword matches as based on word frequencies.*
!	*Prints the instruction for the various Predict/Restore Text Commands.*
\	*Exits the Command Prompt and returns to the main menu.*

Appendix B & C gives examples on how to use Predict/Restore Text Command Prompt to predict and restore text.

Exiting the application

The user may repeatedly select Options 1 till 6. Option 7 is to exit the program.

(*) Take note that for a group of 3 students, there will be two more options to be included for special features. So, Option 7 (to Exit) will then be shifted as option9, the last option.

Requirements for Group Component (70%):

- You are required to design and write the Python application using an Object-Oriented approach (OOP). You should thereby leverage on your knowledge that you have of encapsulation, inheritance, polymorphism etc.
- You may make use of Python's already built in data structures, such as list, tuple, dictionary and set. However, you should refrain from using the classes from the collection library. Instead, you are required to write your own classes to support the various data structures that you may need. Of course, you may refer to the lecture slides and lab tasks and expand further on those classes that we had previously developed in the tutorial and lab sessions.
- To run the application there should be no need to install additional libraries, other than those that ship already with Anaconda.
- The user will be able to start the application from the Anaconda Prompt as follows:

```
python main.py
```

- Your application should not have to rely on any connection to the Internet.
- The OOP classes that you develop must be placed in separate python files.
- The group will be requested to demonstrate the basic features of the application during the demonstration.
- Take note the group's demonstration should not exceed 15 minutes (including 5 minutes for Q&A).

Requirements for Individual Component (30%):

Each individual team member is required to implement two additional features that need to be added to the application as menu options. These two additional features will need to be presented during the final presentation.

- The additional features added must be integrated in the form of additional menu items in the application.
- The features will be graded on technical sophistication and usability.
- Take note, features within the same group must be different. So please check with your group members first before embarking on implementing the extra features.
- Each group member must submit a short PowerPoint deck of slides. Your PowerPoint slides must briefly describe what extra features you have implemented. You must include screen shots demonstrating your extra features in action. Please explain how the features work, and why they are useful. You are to include your Name, Class, and Group Number in the first slide.
- You must submit the PowerPoint Slides (converted to pdf) together with a compulsory Peer Feedback (template will be provided on Brightspace) as well as a duly filled in and signed Academic Integrity Form.

Final Deliverables

Your group's final deliverables must include:

(a) Group Report

A report (as pdf file) with a maximum **of 10** pages. This excludes cover page and the appendix with source listing and references. The report should contain:

a) Cover page with group number, names, ids, and class (your instructor will assign group numbers).
b) Description, and user guidelines, on how to operate your application (please include screen shots of your application in action).
c) Describe how you have made use of the Object-Oriented Programming (OOP) approach. You may elaborate of the classes that you have developed, and discus on issues such as encapsulation, function/operator overloading, polymorphism, inheritance etc. Include a class diagram that displays the relation between the various classes you have developed.
d) Discussion on the data structures and algorithms you have developed for your application. You may discuss issues such as, the performance of the algorithms in terms of Big (O). Explain why you did develop certain data structures and explain why you deem these data structures suitable for the task(s) at hand. Include a table summarizing all the data structures that you have been using (those that you have developed and those already build in Python).
e) Include a summary of the challenges that the group has faced while developing the application (that should include both technical, as well as group-work challenges). Provide a summary of the key takeaways and learning achievements that you have obtained from this project.
f) Include a clear description of the roles and contributions of each member in the team. Clearly state what each member has been responsible for, and what programming work has been carried out by each member.
g) **All** your python **source code** must be included as an appendix at the end of your report. You must clearly indicate in the source code listings who wrote what code. You may include in the appendix, those references from literature or internet that you may have consulted.

(b) Source Code

- You must submit all (*) the python files (.py files) that makes up your application. Ensure code is complete, and it can run directly from the Anaconda Prompt as `python main.py`.

  (*) Take note, that includes the code for all the extra features that were coded by each team member as well.

<u>Submission instructions</u>

Group Submission:

- Group Leader to submit all the group deliverables (Source Code and Group Report) in the designated Brightspace Drop Box.

- Important, the source code <u>must include all the extra features</u> that were coded by the team members. (so, when we run the application, we can experience all the extra features that the team members have developed).

- You must submit it as one Zipped folder (RAR will not be accepted, only zip) whereby you label your submission as:

  CA2_GroupNumberClass.zip

  For example: *CA2_GR_10_DAAA_2B08.zip*

- Please ensure that you submit it by the stipulated deadline.

Individual Submission:

- Each individual Group Member is to submit his/her individual deliverables.

- Individual submission to include:

  - PowerPoint slides describing your two extra features (converted to pdf, maximum 8 slides)
  - Peer Feedback form
  - Academic Integrity Form (filled up & signed)

  Please ensure that you submit it in the designated Brightspace Drop Box for <u>individual submissions</u>.

- You must submit it as one Zipped folder (RAR will not be accepted, only zip) whereby you label your submission as:

  CA2_Final_ GroupNumberStudentNameClass.zip

  For example: *CA2_GR_10_JIMMY_TAN_DAAA_2B08.zip*

- Please ensure to submit it by the stipulated deadline.

Assessment Criteria

The group component of the assignment will be assessed based on the following criteria:

Assessment criteria (Group 70 %)	**Marks awarded**
GROUP COMPONENT (70 %)	
GUI and Command Panel Management: - GUI with main menu that operates as prescribed and has appropriate user input validation and error handling. - Supports two command prompt panels, with instructions and appropriate user input validation and error handling.	Max 10
Basic functionality of the application: Construct/Edit Trie Command Panel: - Can add, delete & find keywords and display trie. - Can write a prefix trie to file, as well as construct a prefix trie from keywords as well as write the keywords from a prefix trie back to file. Predictive/Restore Text Command Panel: - Can list all possible matching keywords for a keyword with wild characters, as well as restore a keyword with wild characters with best match. - Can restore all keywords with wild characters from a text file and restore them using all matching keywords, as well as restore them using best keyword matches.	Max 20
Programming techniques robustness and readability of code: - Appropriate usage of classes and OOP technology. - Appropriate usage of data structures and algorithms. - Code is properly commented and neatly structured. - Application is free of crashes.	Max 20
Group Report: - The report follows the prescribed format. - The report is well written and comprehensive.	Max 10
Group's demonstration: - Group effectively demonstrates the basic features. - Group's ability to answer questions raised in Q&A.	Max 10
Group Total	**70**

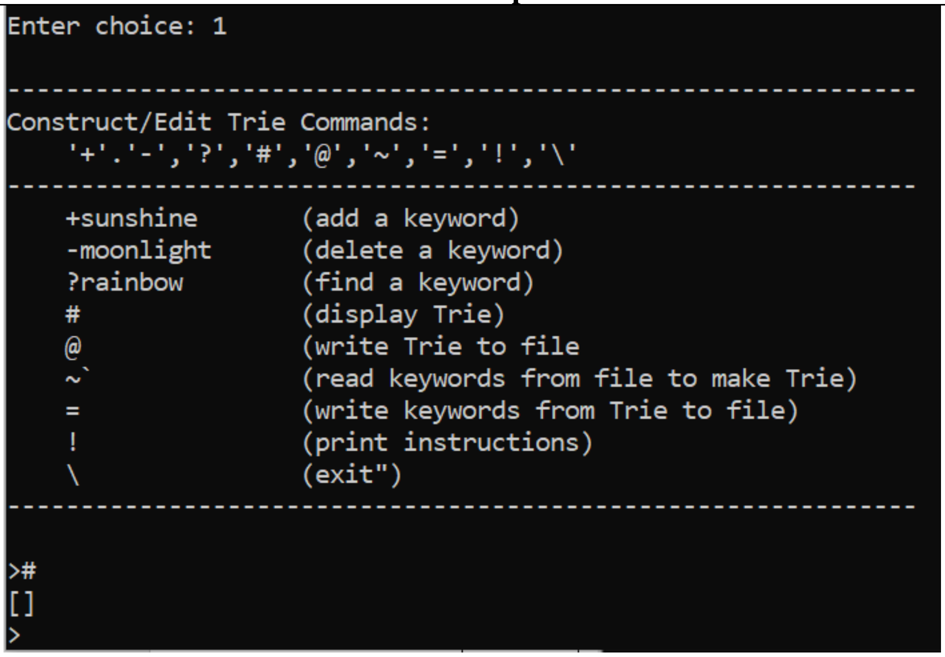
The individual component of the assignment will be assessed based on the following criteria:

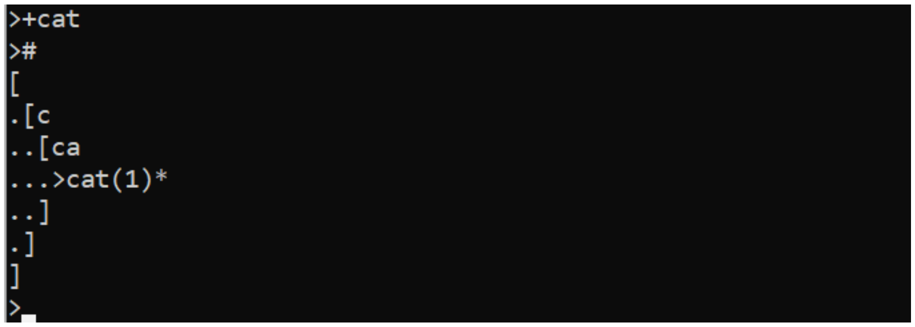
Assessment criteria (Group 70 %)	**Marks awarded**
INDIVIDUAL COMPONENT (30 %)	
Extra Feature One - Technical sophistication. - Usability.	Max 10
Extra Feature Two - Technical sophistication. - Usability.	Max 10
Presentation & Demonstration: - PowerPoint slides. - Demonstration of features and Q&A.	Max 10
Individual Total	**30**

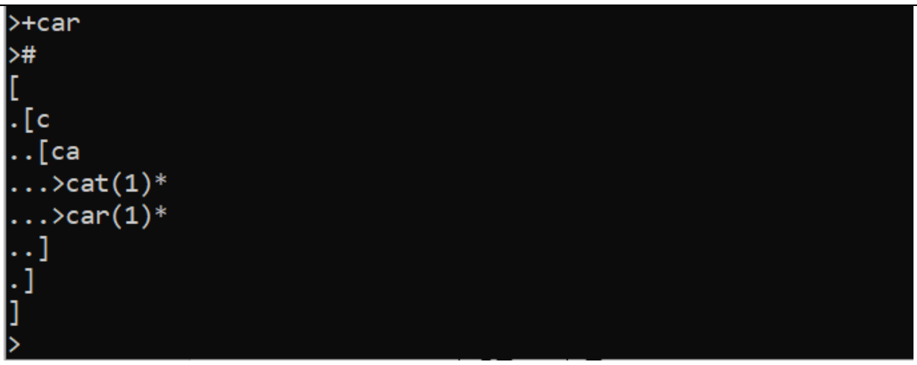
(*) Take note in case of group members with poor contribution to the group effort a multiplier may be applied (peer feedback may thereby be taken into consideration).

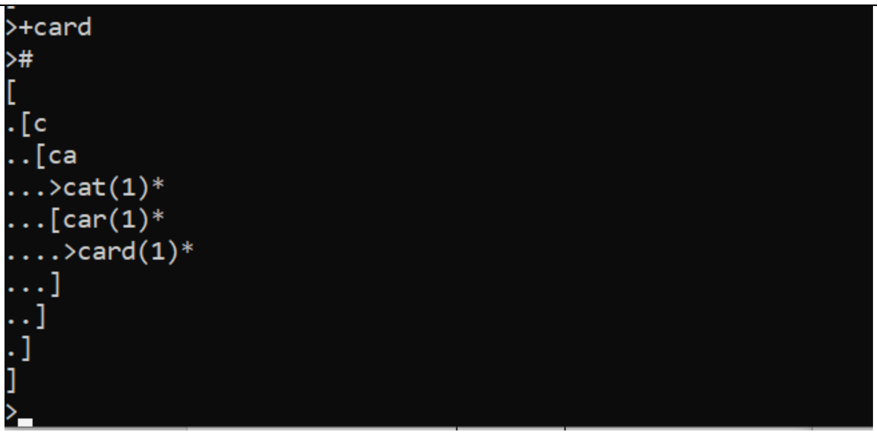
APPENDIX A – Example of a session in Construct/Edit Trie Command Prompt.

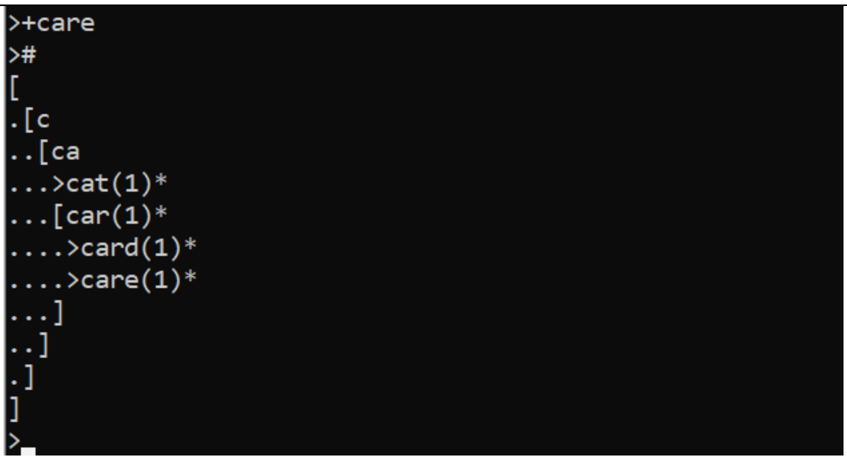
Step	Construct/Edit Trie Command Prompt

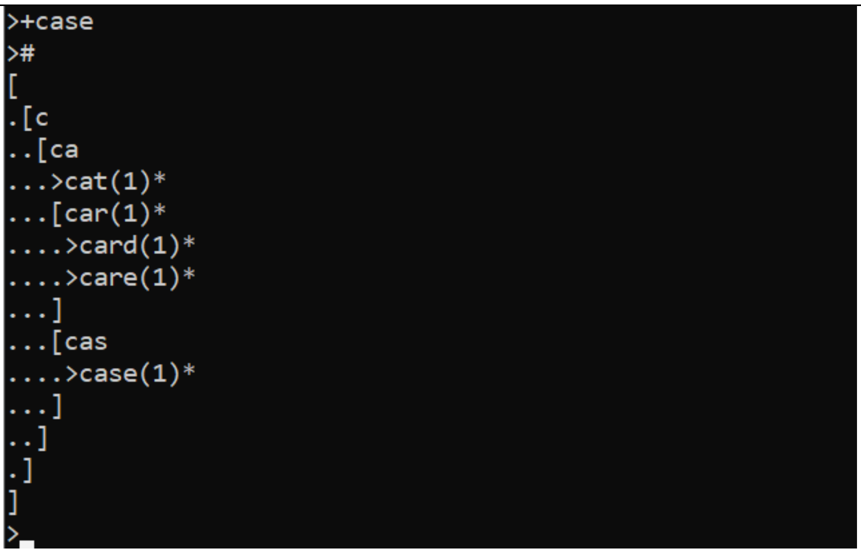
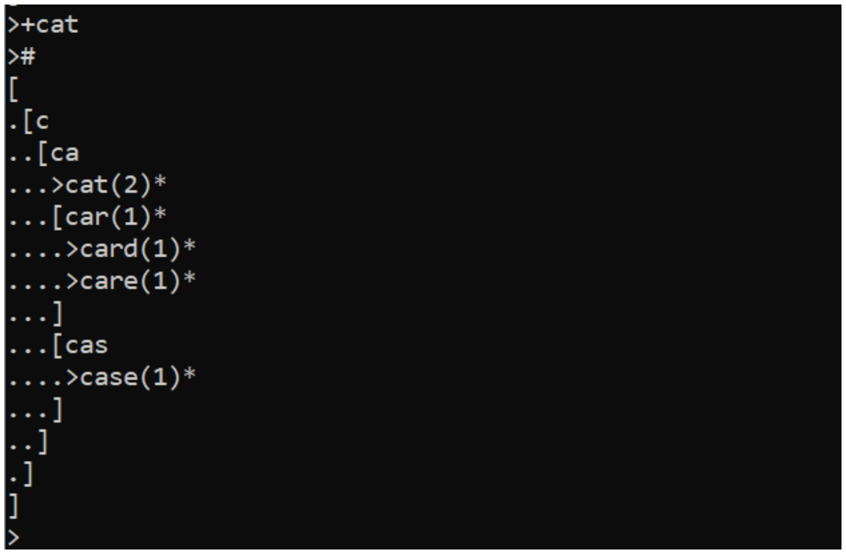
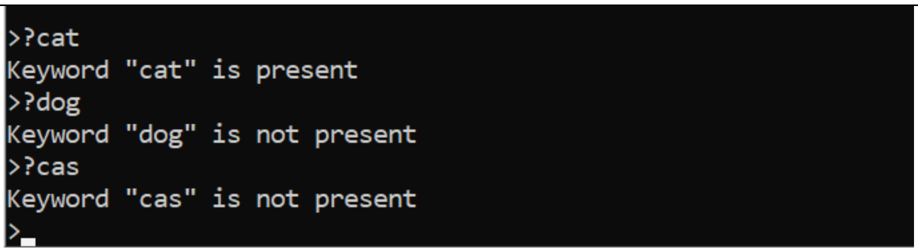
1	 ```Enter choice: 1```
	```--```
	```Construct/Edit Trie Commands:```
	``` '+'.'-','?','#','@','~','=','!','\'```
	```--```
	``` +sunshine (add a keyword)```
	``` -moonlight (delete a keyword)```
	``` ?rainbow (find a keyword)```
	``` # (display Trie)```
	``` @ (write Trie to file```
	``` ~` (read keywords from file to make Trie)```
	``` = (write keywords from Trie to file)```
	``` ! (print instructions)```
	``` \ (exit")```
	```--```
	```>#```
	```[]```
	```>```
	
	When we enter the Edit/Trie Command Prompt we will start with an empty Trie, it will be printed as: []

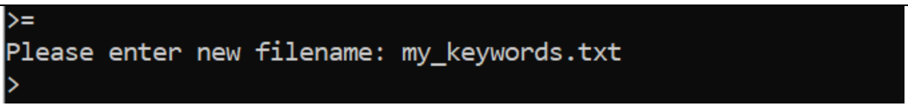
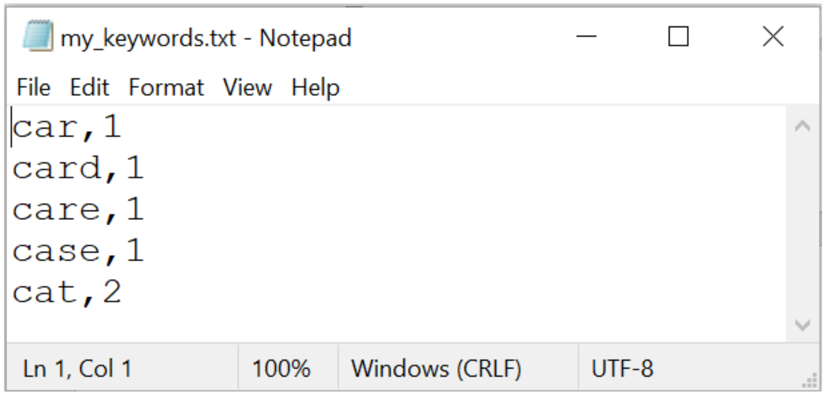
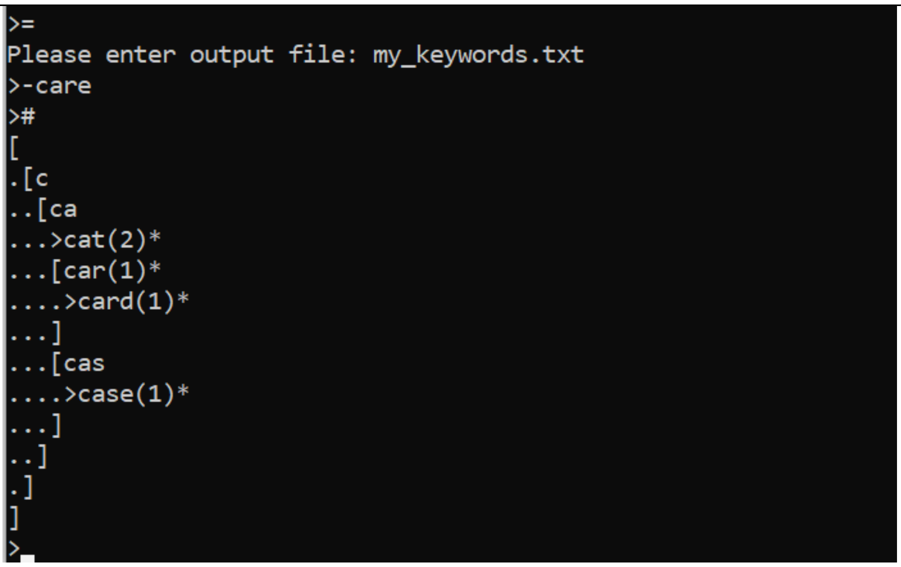
2	 ```>+cat```
	```>#```
	```[```
	```.[c```
	```..[ca```
	```...>cat(1)*```
	```..]```
	```.]```
	```]```
	```>```
	
	Adding a first keyword 'cat' followed by displaying the trie. Notice that only the keywords (terminal nodes) are labeled with an ''. The number in between brackets is the default value (in our case this is a frequency) given to the newly added keyword.*

3	 ```>+car```
	```>#```
	```[```
	```.[c```
	```..[ca```
	```...>cat(1)*```
	```...>car(1)*```
	```..]```
	```.]```
	```]```
	```>```
	Adding a second keyword ‘car’. Notice that ‘cat’ and ‘car’ are sibling nodes with the same parent ‘ca’.

4	 ```>+card```
	```>#```
	```[```
	```.[c```
	```..[ca```
	```...>cat(1)*```
	```...[car(1)*```
	```....>card(1)*```
	```...]```
	```..]```
	```.]```
	```]```
	```>```
	Adding a third keyword ‘card’. Notice that ‘card’ is a child of ‘car’.

5	 ```>+care```
	```>#```
	```[```
	```.[c```
	```..[ca```
	```...>cat(1)*```
	```...[car(1)*```
	```....>card(1)*```
	```....>care(1)*```
	```...]```
	```..]```
	```.]```
	```]```
	```>```
	Adding a fourth keyword ‘care’. Notice that ‘card and ‘care’ are sibling nodes with the same parent ‘car’.

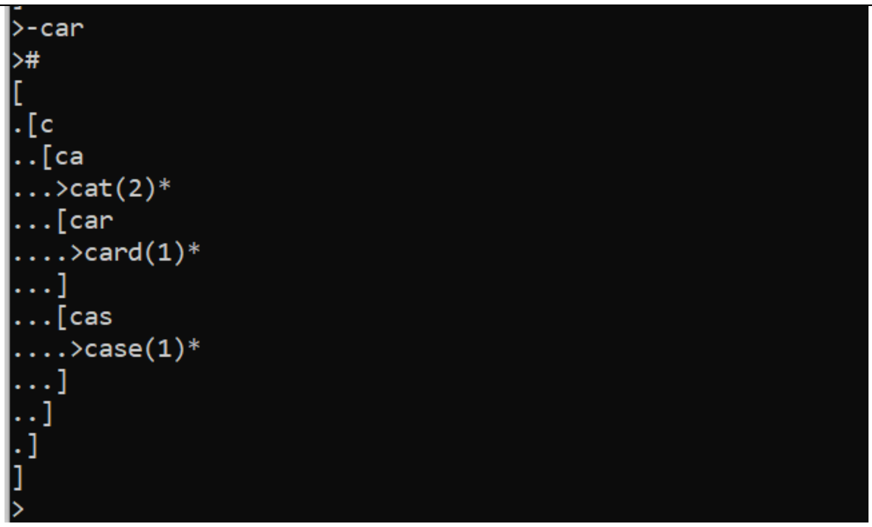
6	 `>+case` `>#` `[` `.[c` `..[ca` `...>cat(1)*` `...[car(1)*` `....>card(1)*` `....>care(1)*` `...]` `...[cas` `....>case(1)*` `...]` `..]` `.]` `]` `>` *Adding a fifth keyword 'case. Notice that 'case' is a child of 'cas'. Keywords, 'cat' and 'car' are siblings of prefix 'cas' as they share the same parent, namely prefix 'ca'.*
7	 `>+cat` `>#` `[` `.[c` `..[ca` `...>cat(2)*` `...[car(1)*` `....>card(1)*` `....>care(1)*` `...]` `...[cas` `....>case(1)*` `...]` `..]` `.]` `]` `>` *Adding keyword 'cat' again, causing the frequency of 'cat' to be increased by one.*
8	 `>?cat` `Keyword "cat" is present` `>?dog` `Keyword "dog" is not present` `>?cas` `Keyword "cas" is not present` `>` *Searching for keywords, notice only keyword 'cat' is present. Take note 'cas' is not a keyword but a prefix.*

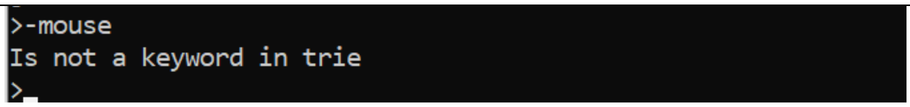
8	 `>=` `Please enter new filename: my_keywords.txt` `>`  `car,1` `card,1` `care,1` `case,1` `cat,2` *Keywords can be extracted from the current trie and then stored in a text file.*
9	 *Example of deleting a keyword that has no children (e.g. is a leaf node), in this case the keyword ‘care’ is being deleted. Notice that ‘care’ does not show up when we display the trie again.*

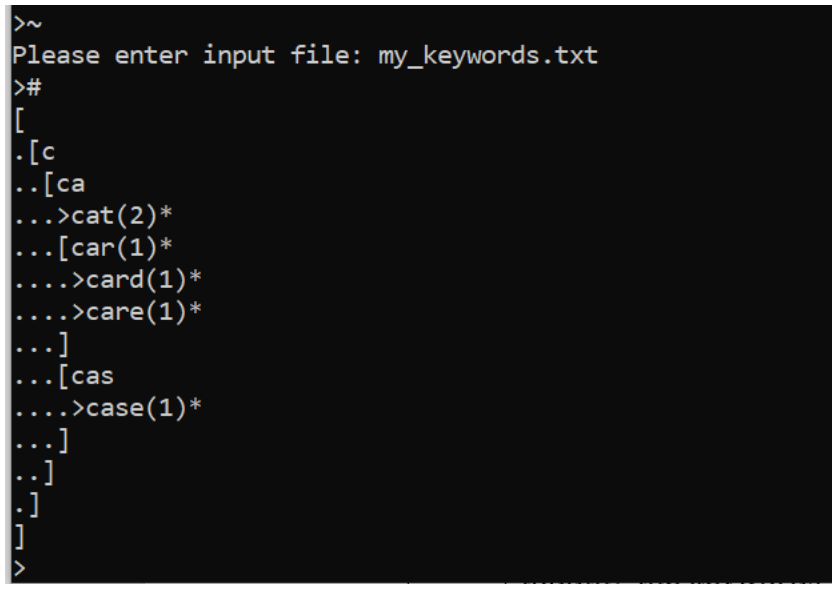
```
>=
Please enter new filename: my_keywords.txt
>
```

```
car,1
card,1
care,1
case,1
cat,2
```

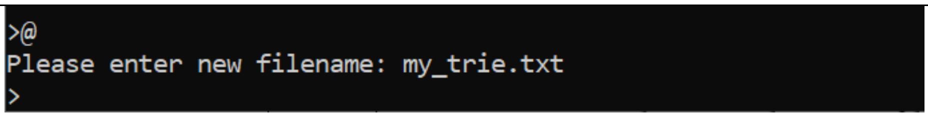
```
>=
Please enter output file: my_keywords.txt
>-care
>#
[
.[c
..[ca
...>cat(2)*
...[car(1)*
....>card(1)*
...]
...[cas
....>case(1)*
...]
..]
.]
]
>
```

10	
	```
	>-car
	>#
	[
	.[c
	..[ca
	...>cat(2)*
	...[car
	....>card(1)*
	...]
	...[cas
	....>case(1)*
	...]
	..]
	.]
	]
	>
	```
	 Example of deleting a keyword that has children (is not a leaf node), in this case the keyword 'car' is being deleted. Notice that 'car still shows up when we display the trie again. However, 'car' has now been demoted as a prefix, and is not a keyword anymore.

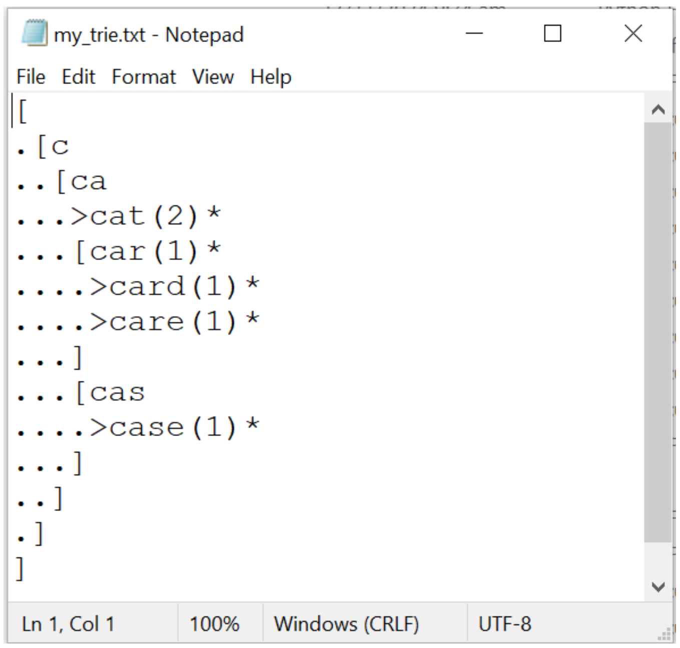
11	
	```
	>-mouse
	Is not a keyword in trie
	>_
	```
	 An attempt to delete a word that is not a keyword will produce the above warning message.

12	
	```
	>~
	Please enter input file: my_keywords.txt
	>#
	[
	.[c
	..[ca
	...>cat(2)*
	...[car(1)*
	....>card(1)*
	....>care(1)*
	...]
	...[cas
	....>case(1)*
	...]
	..]
	.]
	]
	>
	```
	 After deleting some of the keywords we can restore the original trie again by loading the keywords file that we had earlier on saved in a file . Take note that by reading in the keywords we will clear the current trie, and we repopulate it using the keywords from the file.

13



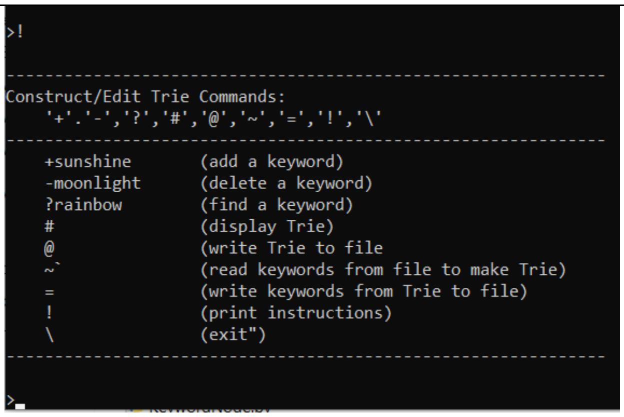
```
>@
Please enter new filename: my_trie.txt
>
```



```
[
.[c
..[ca
...>cat(2)*
...[car(1)*
....>card(1)*
....>care(1)*
...]
...[cas
....>case(1)*
...]
..]
.]
]
```

You may write the trie also to a file, similarly as you display it on the screen.

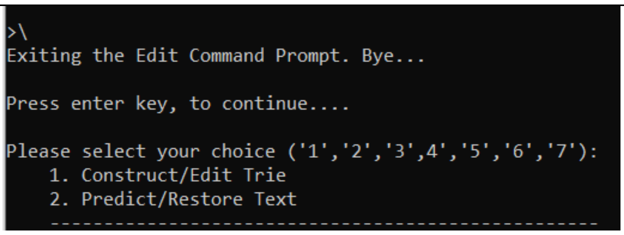
14



```
>!

----------------------------------------------------------------
Construct/Edit Trie Commands:
    '+'.'-','?','#','@','~','=','!','\'
----------------------------------------------------------------
    +sunshine       (add a keyword)
    -moonlight      (delete a keyword)
    ?rainbow        (find a keyword)
    #               (display Trie)
    @               (write Trie to file
    ~`              (read keywords from file to make Trie)
    =               (write keywords from Trie to file)
    !               (print instructions)
    \               (exit")
----------------------------------------------------------------

>
```

Printing the instructions.

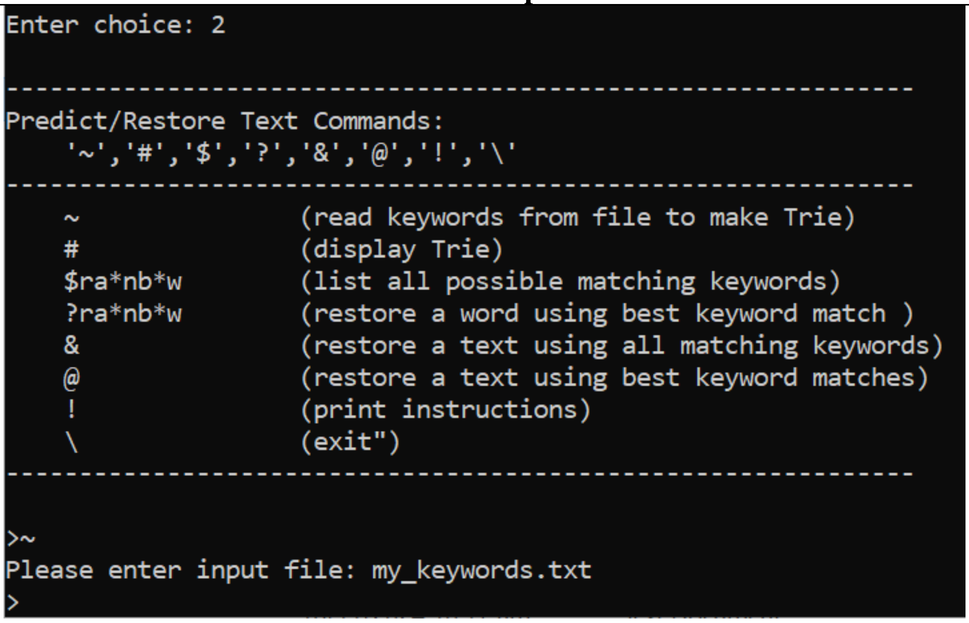
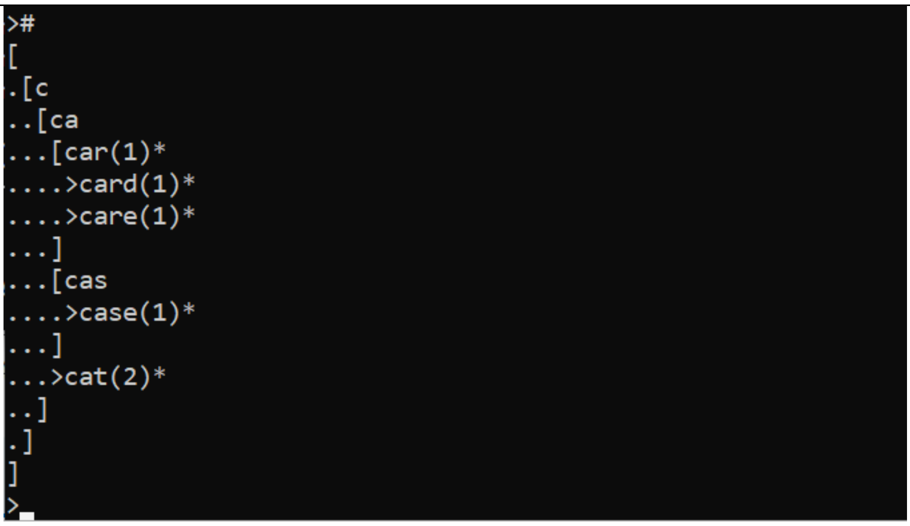
15

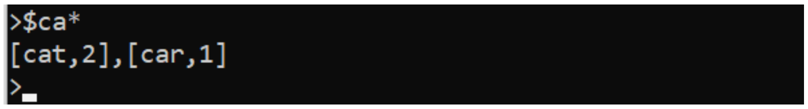
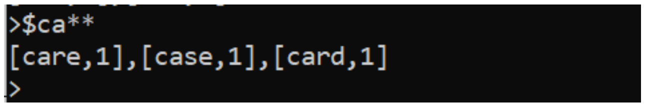
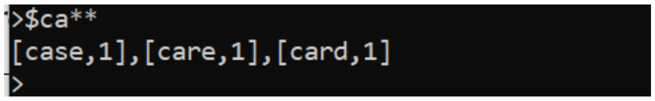
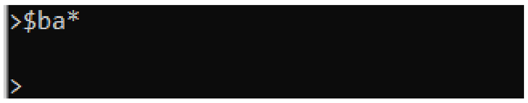


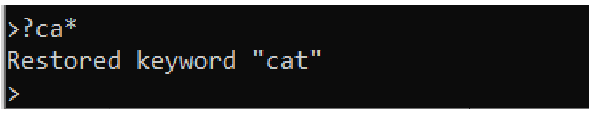
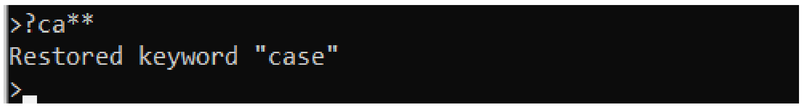
```
>\
Exiting the Edit Command Prompt. Bye...

Press enter key, to continue....

Please select your choice ('1','2','3',4','5','6','7'):
    1. Construct/Edit Trie
    2. Predict/Restore Text
    ------------------------------------------------
```

Exiting the command prompt

APPENDIX B – Example of a session in Predict/Restore Text Command Prompt.

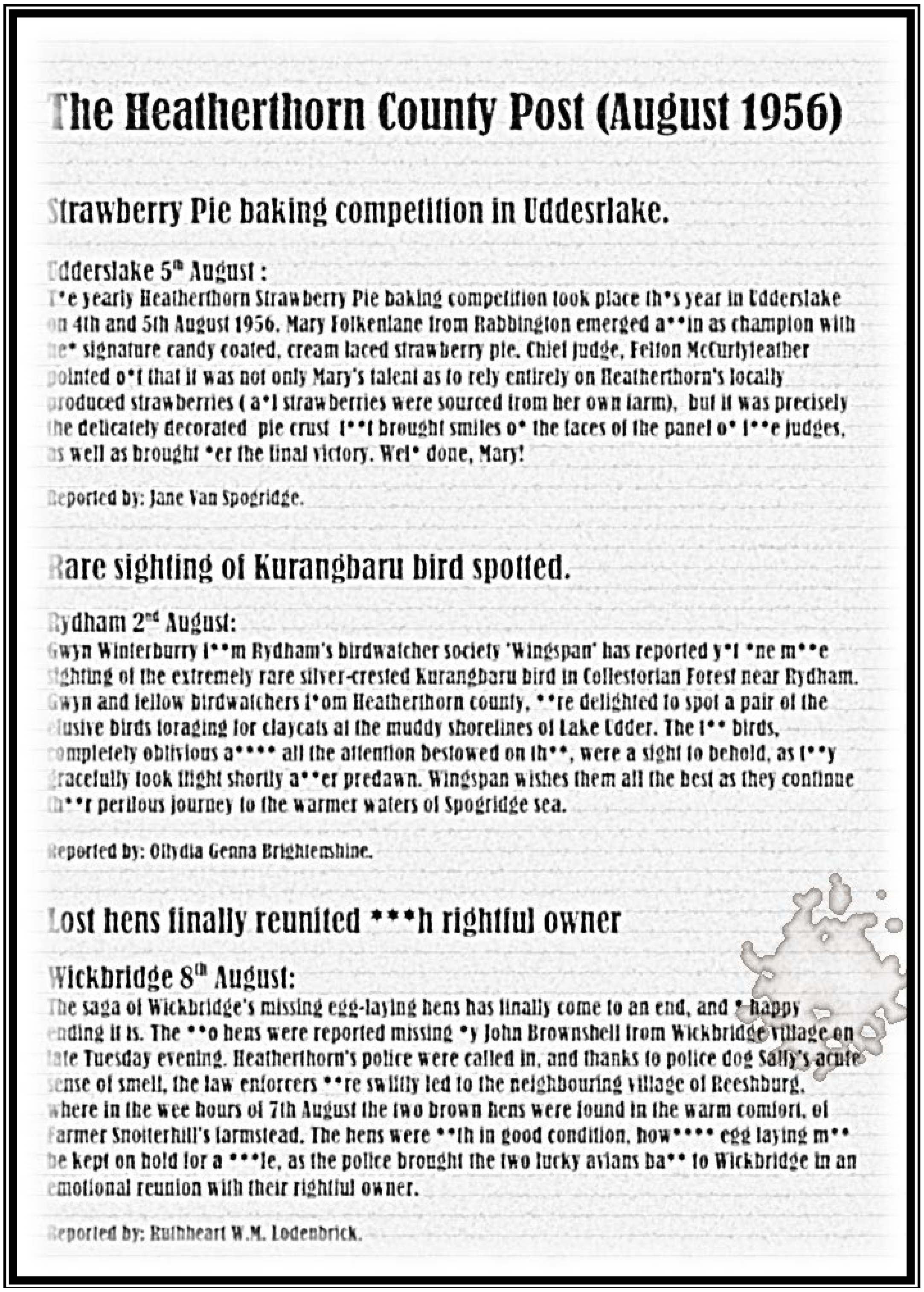
Step	Predict/Restore Text Command Prompt
1	 ```Enter choice: 2``` ```---``` ```Predict/Restore Text Commands:``` ``` '~','#','$','?','&','@','!','\'``` ```---``` ``` ~ (read keywords from file to make Trie)``` ``` # (display Trie)``` ``` $ra*nb*w (list all possible matching keywords)``` ``` ?ra*nb*w (restore a word using best keyword match)``` ``` & (restore a text using all matching keywords)``` ``` @ (restore a text using best keyword matches)``` ``` ! (print instructions)``` ``` \ (exit")``` ```---``` ```>~``` ```Please enter input file: my_keywords.txt``` ```>``` *When we enter the Predict/Restore Command prompt we may start off by loading a trie from a keyword file.*
2	 ```>#``` ```[``` ```.[c``` ```..[ca``` ```...[car(1)*``` ```....>card(1)*``` ```....>care(1)*``` ```...]``` ```...[cas``` ```....>case(1)*``` ```...]``` ```...>cat(2)*``` ```..]``` ```.]``` ```]``` ```>``` *Note that the trie we loaded from the file contains the keywords: 'car', 'card', 'care', 'case' and 'cat'.*

3	
	`>$ca*`
	`[cat,2],[car,1]`
	`>`
	
	We may use the wild card character '' to list all possible matches with keywords in the trie. If there are multiple matches, the keywords that are being returned will be sorted with those with higher frequencies displayed first (so keyword 'cat' precedes keyword 'car' as it has a higher frequency).*
	
	
	`>$ca**`
	`[care,1],[case,1],[card,1]`
	`>`
	
	Next another example of listing all the possible matches for a word with wild card characters with keywords in the trie. Take note that for those keywords with same frequencies they will be displayed in random order, so would you run this command again the order of 'case', 'care', and 'card' may be different as they all have the same frequency.
	
	
	`>$ca**`
	`[case,1],[care,1],[card,1]`
	`>`
	
	If there are no matches for a word with wild card characters then an empty line will be displayed.
	
	
	`>$ba*`
	
	`>`

4	
	`>?ca*`
	`Restored keyword "cat"`
	`>`
	
	To restore a keyword with wild card characters using a best match (e.g. use the matching word with highest frequency) use the '?' command.
	
	
	`>?ca**`
	`Restored keyword "case"`
	`>`
	
	Another example of restoring a keyword with wild card characters. Take note that for those best matching keywords with same frequencies they will be picked in random order, so would you run this command again choice between 'case', 'care', and 'card' may be different as they all have the same frequency.

APPENDIX C – Restoring the Heatherthorn County Post (August 1956)

A page from the Heatherthorn County Post containing three news clips that need restoration.



The Heatherthorn County Post (August 1956)

Strawberry Pie baking competition in Uddesrlake.

Udderslake 5th August :
T*e yearly Heatherthorn Strawberry Pie baking competition took place th*s year in Udderslake on 4th and 5th August 1956. Mary Folkenlane from Rabbington emerged a**in as champion with he* signature candy coated, cream laced strawberry pie. Chief judge, Felton McCurlyfeather pointed o*t that it was not only Mary's talent as to rely entirely on Heatherthorn's locally produced strawberries (a*l strawberries were sourced from her own farm), but it was precisely the delicately decorated pie crust t**t brought smiles o* the faces of the panel o* t**e judges, as well as brought *er the final victory. Wel* done, Mary!

Reported by: Jane Van Spogridge.

Rare sighting of Kurangbaru bird spotted.

Rydham 2nd August:
Gwyn Winterburry f**m Rydham's birdwatcher society 'Wingspan' has reported y*t *ne m**e sighting of the extremely rare silver-crested Kurangbaru bird in Collestorian Forest near Rydham. Gwyn and fellow birdwatchers f*om Heatherthorn county, **re delighted to spot a pair of the elusive birds foraging for claycats at the muddy shorelines of Lake Udder. The t** birds, completely oblivious a**** all the attention bestowed on th**, were a sight to behold, as t**y gracefully took flight shortly a**er predawn. Wingspan wishes them all the best as they continue th**r perilous journey to the warmer waters of Spogridge sea.

Reported by: Ollydia Genna Brightenshine.

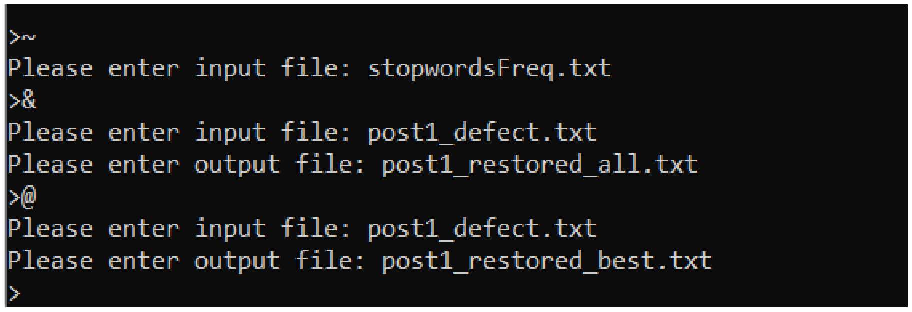
Lost hens finally reunited ***h rightful owner

Wickbridge 8th August:
The saga of Wickbridge's missing egg-laying hens has finally come to an end, and * happy ending it is. The **o hens were reported missing *y John Brownshell from Wickbridge village on late Tuesday evening. Heatherthorn's police were called in, and thanks to police dog Sally's acute sense of smell, the law enforcers **re swiftly led to the neighbouring village of Reeshburg, where in the wee hours of 7th August the two brown hens were found in the warm comfort, of Farmer Snotterhill's farmstead. The hens were **th in good condition, how**** egg laying m** be kept on hold for a ***le, as the police brought the two lucky avians ba** to Wickbridge in an emotional reunion with their rightful owner.

Reported by: Ruthheart W.M. Lodenbrick.

(*) Take note: As for restoring the three clips a prefix trie was used that was constructed out of a list of keywords (stopwords) and associated frequencies that you can download from Brightspace (filename: 'stopwordsFreq.txt')

Example 1: Heatherthorn Post news clip one



```
>~
Please enter input file: stopwordsFreq.txt
>&
Please enter input file: post1_defect.txt
Please enter output file: post1_restored_all.txt
>@
Please enter input file: post1_defect.txt
Please enter output file: post1_restored_best.txt
>
```

In this case we read the keywords to build a prefix trie from a file 'stopwordsFreq.txt'. Thereafter we restore a news clip with wild card characters (post1_defect.txt') twice. First by including all possible matches, second by including only the best matches.

Post 1: post1_defect.txt

Strawberry Pie baking competition in Udderslake.

Udderslake 5th August:
T*e yearly Heatherthorn Strawberry Pie baking competition took place th*s year in Udderslake on 4th and 5th August 1956. Mary Folkenlane from Rabbington emerged a**in as champion with he* signature candy coated, cream laced strawberry pie. Chief judge, Felton McCurlyfeather pointed o*t that it was not only Mary's talent as to rely entirely on Heatherthorn's locally produced strawberries (a*l strawberries were sourced from her own farm), but it was precisely the delicately decorated pie crust t**t brought smiles o* the faces of the panel o* f**e judges, as well as brought *er the final victory. Wel* done, Mary!

Reported by: Jane Van Spogridge.

Post 1: post1_restored_all.txt

Strawberry Pie baking competition in Udderslake.

Udderslake 5th August:
['the'] yearly Heatherthorn Strawberry Pie baking competition took place ['this','thus'] year in Udderslake on 4th and 5th August 1956. Mary Folkenlane from Rabbington emerged ['again'] as champion with ['her'] signature candy coated, cream laced strawberry pie. Chief judge, Felton McCurlyfeather pointed ['out'] that it was not only Mary's talent as to rely entirely on Heatherthorn's locally produced strawberries (['all'] strawberries were sourced from her own farm), but it was precisely the delicately decorated pie crust ['that'] brought smiles ['of','on','or'] the faces of the panel ['of','on','or'] ['five','fire'] judges, as well as brought ['her','per'] the final victory. ['well'] done, Mary!

Reported by: Jane Van Spogridge.

Post 1: post1_restored_best.txt

Strawberry Pie baking competition in Udderslake.

Udderslake 5th August:
<The> yearly Heatherthorn Strawberry Pie baking competition took place <this> year in Udderslake on 4th and 5th August 1956. Mary Folkenlane from Rabbington emerged <again> as champion with <her> signature candy coated, cream laced strawberry pie. Chief judge, Felton McCurlyfeather pointed <out> that it was not only Mary's talent as to rely entirely on Heatherthorn's locally produced strawberries (<all> strawberries were sourced from her own farm), but it was precisely the delicately decorated pie crust <that> brought smiles <of> the faces of the panel <of> <five> judges, as well as brought <her> the final victory. <Well> done, Mary!

Reported by: Jane Van Spogridge.

Example 2: Heatherthorn Post news clip two

Post 2: post2_defect.txt

Rare sighting of Kurangbaru bird spotted.

Rydham 2nd August:
Gwyn Winterburry f**m Rydham's birdwatcher society 'Wingspan' has reported y*t *ne m**e sighting of the extremely rare silver-crested Kurangbaru bird in Collestorian Forest near Rydham. Gwyn and fellow birdwatchers f*om Heatherthorn county, **re delighted to spot a pair of the elusive birds foraging for claycats at the muddy shorelines of Lake Udder. The t** birds, completely oblivious a**** all the attention bestowed on th**, were a sight to behold, as t**y gracefully took flight shortly a**er predawn. Wingspan wishes them all the best as they continue th**r perilous journey to the warmer waters of Spogridge sea.

Reported by: Ollydia Genna Brightenshine.

Post 2: post2_restored_all.txt

Rare sighting of Kurangbaru bird spotted.

Rydham 2nd August:
Gwyn Winterburry ['from'] Rydham's birdwatcher society'Wingspan' has reported ['yet'] ['one'] ['more','made','move','mine'] sighting of the extremely rare silver-crested Kurangbaru bird in Collestorian Forest near Rydham. Gwyn and fellow birdwatchers ['from'] Heatherthorn county, ['more','here','were','fire'] delighted to spot a pair of the elusive birds foraging for claycats at the muddy shorelines of Lake Udder. The ['the','top','two','too','ten'] birds, completely oblivious ['about','after','again','above','along','am ong','alone'] all the attention bestowed on ['that','this','they','than','them','then','thus','thin','thru'], were a sight to behold, as ['they'] gracefully took flight shortly ['after'] predawn. Wingspan wishes them all the best as they continue ['their'] perilous journey to the warmer waters of Spogridge sea.

Reported by: Ollydia Genna Brightenshine.

Post 2: post2_restored_best.txt

Rare sighting of Kurangbaru bird spotted.

Rydham 2nd August:
Gwyn Winterburry <from> Rydham's birdwatcher society'Wingspan' has reported <yet> <one> <more> sighting of the extremely rare silver-crested Kurangbaru bird in Collestorian Forest near Rydham. Gwyn and fellow birdwatchers <from> Heatherthorn county, <more> delighted to spot a pair of the elusive birds foraging for claycats at the muddy shorelines of Lake Udder. The <the> birds, completely oblivious <about> all the attention bestowed on <that>, were a sight to behold, as <they> gracefully took flight shortly <after> predawn. Wingspan wishes them all the best as they continue <their> perilous journey to the warmer waters of Spogridge sea.

Reported by: Ollydia Genna Brightenshine.

Example 3: Heatherthorn Post news clip three

Post 3: post3_defect.txt

Lost hens finally reunited ***h rightful owner

Wickbridge 8th August:
The saga of Wickbridge's missing egg-laying hens has finally come to an end, and * happy ending it is. The **o hens were reported missing *y John Brownshell from Wickbridge village on late Tuesday evening. Heatherthorn's police were called in, and thanks to police dog Sally's acute sense of smell, the law enforcers **re swiftly led to the neighbouring village of Reeshburg, where in the wee hours of 7th August the two brown hens were found in the warm comfort, of Farmer Snoiterhill's farmstead. The hens were **th in good condition, how**** egg laying m** be kept on hold for a ***le, as the police brought the two lucky avians ba** to Wickbridge in an emotional reunion with their rightful owner.

Reported by: Ruthheart W.M. Lodenbrick

Post 3: post3_restored_all.txt

Lost hens finally reunited ['with','such','each','much','both'] rightful owner

Wickbridge 8th August:
The saga of Wickbridge's missing egg-laying hens has finally come to an end, and ['a','i'] happy ending it is. The ['who','two','too'] hens were reported missing ['by','my'] John Brownshell from Wickbridge village on late Tuesday evening. Heatherthorn's police were called in, and thanks to police dog Sally's acute sense of smell, the law enforcers ['more','here','were','fire'] swiftly led to the neighbouring village of Reeshburg, where in the wee hours of 7th August the two brown hens were found in the warm comfort, of Farmer Snoiterhill's farmstead. The hens were ['with','both'] in good condition, ['however'] egg laying ['may'] be kept on hold for a ['while','whole'], as the police brought the two lucky avians ['back'] to Wickbridge in an emotional reunion with their rightful owner.

Reported by: Ruthheart W.M. Lodenbrick

Post 3: post_3_restored_best

Lost hens finally reunited <with> rightful owner

Wickbridge 8th August:
The saga of Wickbridge's missing egg-laying hens has finally come to an end, and <a> happy ending it is. The <who> hens were reported missing <by> John Brownshell from Wickbridge village on late Tuesday evening. Heatherthorn's police were called in, and thanks to police dog Sally's acute sense of smell, the law enforcers <more> swiftly led to the neighbouring village of Reeshburg, where in the wee hours of 7th August the two brown hens were found in the warm comfort, of Farmer Snoiterhill's farmstead. The hens were <with> in good condition, <however> egg laying <may> be kept on hold for a <while>, as the police brought the two lucky avians <back> to Wickbridge in an emotional reunion with their rightful owner.

Reported by: Ruthheart W.M. Lodenbrick

~ End ~