CITS 2401
Computer Analysis and Visualisation

THE UNIVERSITY OF WESTERN AUSTRALIA

Assignment 2
Hajana

Worth: 3% of the unit

Submission: Answer the questions on the quiz server.
Deadline: 3 April 2020 5pm
Late submissions: late submissions attract 5% penalty per day up to 7 days (i.e., 10 April 2020 5pm). After, the mark will be 0 (zero). Also, any plagiarised work will be marked zero.

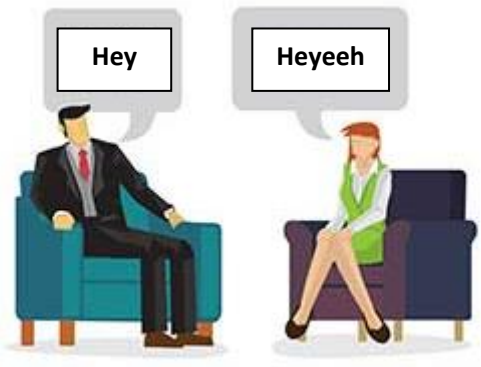
1. Outline

Hajana*, a land of small country located in the pacific, speaks their own language Hajanian. This isolated country was ruled by the British Empire a long time ago, hence their language is highly influenced by English. They have adopted English, but as many years have passed, the way they use English has changed. When an explorer visited Hajana back, the explorer have noted the following rules that has changed from the UK English rule.

1. All words ending in "y" has been converted to "yeeh". E.g., "Company" -> "Companyeeh".
2. All words ending in "ing" has been converted to "ingah". E.g., "bring" -> "bringah".
3. All words ending in "sh" has been converted to "shey". E.g., "wish" -> "wishey".
4. All words with "oo" within the word (including beginning and ending) has been converted to "uwu". E.g., "Boolean" -> "Buwulean", "shoo" -> "shuwu".
5. The word "is" has been changed to "oy".
6. The word "are" has been changed to "argh".
7. All words "he", "she" and "it" are changed to "hana".
8. All words "his" and "her" are changed to "hami".

Finally, their favourite food "ham", "sausage" and "bacon" are all added with an exclamation mark whenever in writing. E.g., "ham" -> "ham!".

In this assignment, we will develop an interpreter for us to communicate with Hajanians. All the tasks can be answered with materials up to loops.



*Note, names, places and references used in this assignment has no relation to them in real world.

CITS 2401
Computer Analysis and Visualisation

2. Tasks

Task 1: English to Hajanian word
Write a function `english_word_to_hajanian(word)` that converts a given English word `word` (type string) into a Hajanian word following the rules given above. You can assume the English word given is always in lowercase.

Task 2: Hajanian to English word
Write a function `hajanian_word_to_english(word)` that converts a given Hajanian word `word` (type string) into an English word following the rules given above. Because words "hana" and "hami" could be converted from multiple English words, for simplicity we will convert to "he" and "his", respectively (e.g., all "hana" words are converted to "he"). You can assume the Hajanian word given is always in lowercase.

Task 3: Valid word filtering
An input sentence by a user may contains typos or punctuations, which would cause a lot of error in translations. To overcome this problem, the first task is to create a dictionary and filter invalid words. Write a function `valid_words(dictionary, sentence)` which takes in a string of English words `dictionary` (type string) and `sentence` (type string), and returns a list of words (type list) that filters invalid words in `sentence` (i.e., words not in `dictionary`).

For example, `dictionary` string "a aa apple banana pear strawberry mango" and `sentence` string "hello mango and banana yum apple!" would result in a list of strings ["mango", "banana"]. Note that "apple!" isn't matched as a valid word due to the exclamation mark being part of the word.

Task 4: Beautify sentence
As you would have noticed in Task 3 above, punctuations can lead to filtering valid words (e.g., "apple!" for example above). In this task, write a function `beautify_sentence(sentence, punctuation)` which returns a new sentence that removes all the specified punctuations from words (words are separated by white space).

For example, `sentence` "?hello !mango! and, ban,ana yum apple!" with `punctuation` "?!," would result in returning a string "hello mango and ban,ana yum apple". Notice that "ban,ana" still contains comma.

Task 5: Translating sentences
This task translates English to Hajanian, and Hajanian to English using functions developed in above tasks. So, write a function `translate(sentence, dictionary, punctuation, language)` that translates the given `sentence` to the other language specified in `language` (e.g., if `language` is "english", then translate `sentence` to Hajanian, and vice versa). You should remember to beautify the sentence by removing punctuations specified in `punctuation`. Finally, reconstruct the sentence of translated sentence. If an invalid language is given, the function prints "invalid language." and returns an empty string (i.e., "").

For example, we have the following:
```
sentence = "?hello boolean bring, !mango! and, country ban,ana wish yum apple!"
dictionary = "a aa apple banana pear strawberry mango country wish boolean bring"
punctuation = "?!,"
language = "english"
```

This would return "buwulean bringah mango countryeeh wishey apple"

Please note, you are expected to use functions implemented in the previous tasks.

CITS 2401
Computer Analysis and Visualisation

Task 6: User interaction

To enable interaction with users, write a function `user_interaction()` that does the following:

1. Prompt to user "Which language?", where the user type in "english" or "hajanian" (without the double quotes).
2. Prompt to user "Do you have a dictionary?", where the user type in "yes" or "no". If "yes", go to step 3, else go to step 4.
3. Prompt to user "Enter dictionary sentence.", where the user will enter the dictionary sentence.
4. Prompt to user "Enter punctuation filters.", where the user will enter the punctuations to filter.
5. Prompt to user "Enter [language] sentence.", where [language] is replaced with the language chosen in step 1, and the user enters the sentence to convert.
6. Finally, the function prints the converted sentence (print to the terminal) and finishes.

Please note the following scenarios:

A. If dictionary and punctuation filters are not specified (i.e., an empty string is entered for both), then consider all words in the sentence as valid (think about achieving this without modifying existing functions).
B. If dictionary is not specified but punctuation filters are specified, (i.e., an empty string for dictionary, and non-empty string for punctuation filter), then consider all words as valid AFTER filtering punctuations.
C. If dictionary is specified but punctuation filters are not specified (i.e., non-empty string for dictionary, and an empty string for punctuation filters), then valid words are only ones specified in the dictionary.
D. If both dictionary and punctuation filters are specified, then apply the punctuation filter and then find valid words using the dictionary.

Example scenario 1:
Which language? english
Do you have a dictionary? yes
Enter dictionary sentence. a aa apple banana pear strawberry mango country wish boolean bring
Enter punctuation filters. ?!,
Enter english sentence. ?hello boolean bring, !mango! and, country ban,ana wish yum apple!
buwulean bringah mango countryeeh wishey apple

Example scenario 2:
Which language? hajanian
Do you have a dictionary? no
Enter punctuation filters.
Enter hajanian sentence. this oy has wishey washey thingah hana ham! bacon! so yum uwulong guwud.
this is has wish wash thing he ham bacon so yum oolong good.

(note, user inputs are coloured in blue in the above examples, all others are program generated)

Note 1: You can assume that the user will always input valid commands.
Note 2: You should be using functions you have implemented in the previous tasks.
Note 3: Remember, there is white space after the user prompt.

CITS 2401
Computer Analysis and Visualisation

Task 7: Calculating the interpretation error rate
Implement a function `error_rate(sentence)` that takes in an English sentence, and calculate the error in percentage that the implemented interpreter makes with the given input `sentence`. The error is calculated by converting the English sentence (version 1, the original input `sentence`) into Hajanian, and then converted back to English (version 2). Then, the two versions are compared word by word (words are separated by white space) to calculate the percentage. The two sentences below shows the error calculation:

Version 1: "she was going to give me some time but her schedule said otherwise."
Version 2: "he was going to give me some time but his schedule said otherwise."

Then, we get the following statistics:
word count of version 1: 13
Different words: 2
Error rate: 2/13 = 0.15384615384615385

Here, you can assume there are no dictionary or punctuation filters (i.e., same as Task 6 scenario A).

Your function should return **only the fraction value** (e.g., 0.15384615384615385 for the above example).

3. Submission

You should answer the questions related to the tasks above on the quiz server by the due date - 3 April 2020 5pm (drop dead due date 10 April 2020 with 5% penalty per day).