FIT1045 Algorithmic Problem Solving – Workshop 5.

Objectives

The **objectives of this workshop** are:

- To practice string manipulations
- To implement algorithms to solve large tasks involving strings
- To develop skills and understanding of practical uses of tables

Useful Material

Reading from files: Section 7.2.1
`https://docs.python.org/3/tutorial/inputoutput.html#reading-and-writing-files`

Lists: `https://docs.python.org/3/tutorial/datastructures.html`

Task 1:

Part 1 A palindrome is a word that is spelt the same forward and backwards. Your task is to write a program that takes, as input, a single word and returns yes if it is a palindrome and no otherwise.

Example:

```
Please enter a word: dendrite
Unfortunately dendrite is not a palindrome.

Please enter a word: aibohphobia
Great, aibohphobia is a palindrome.
```

Note: If you have spare time, google the definition of aibohphobia.

Part 2 Pure palindromic sentences in English are often difficult to find, this is due to the structure of English grammar. As such we often loosen the constraint surrounding sentences and allow certain grammatical structures to be ignored. Write a program that ignores spaces and the following punctuation [“,", “?", “."] and determines if the sentence is a palindrome.

For instance, “Live on evasions? No, I save no evil." is a palindrome if you ignore the space character as well as the following punctuation [“,", “?", “."].

Example:

```
Please enter a sentence: Live on evasions? No, I save no evil.
This sentence is a palindrome.

Please enter a word: A quick brown fox jumps over the lazy dog.
This sentence is not a palindrome.
```

Task 2:

The file, 'palindromic.txt', contains a form of palindrome based crossword. Write a program that reads this file into a table where each element contains a single character, once you have read the file print a list of palondromes that can be found in the table considering both rows and columns. The ouput for your program should have both the starting index of the palindrome and the direction of the palindrome.

Sample output:

```
rotor is a palindrome starting at [0][0] and is a row in the table
refer is a palindrome starting at [0][0] and is a column in the table
tenet is a palindrome starting at [0][2] and is a column in the table
radar is a palindrome starting at [0][4] and is a column in the table
```

Task 3:

A $N \times N$ matrix is a table of elements arranged in N rows and N columns. Given a matrix, M, we are able to take the transpose of a matrix M^T by taking the rows of the matrix M and making them the columns of the matrix M^T.

$$\begin{pmatrix} a & b & c \\ d & e & f \\ g & h & i \end{pmatrix} \rightarrow \begin{pmatrix} a & d & g \\ b & e & h \\ c & f & i \end{pmatrix}$$

Write a program that reads the file "matrix.txt", stores it as a table of integers and creates a second table to contain the transpose. Once you have created the transpose be sure to print the transpose and the original matrix for comparison. Once you have tested your program try to run it on "matrix_large.txt"

Task 4:

Modify your program from the previous task so that you create the transpose without using an additional table.

Hint: Try using temporary variables so that you can freely swap elements.

Task 5:

Write a program that removes duplicate characters from a string, leaving only the first occurence of that character.

Example:

```
Please enter a word: Bananas
The duplicate free string from the word Bananas is Bans.
```

Note: "Bnas" would be an incorrect string, whilst it has all the correct symbols it does not conserve the order in which they appear.

Extend your program to keep track of the number of occurences of the letters.