### TASK 1:

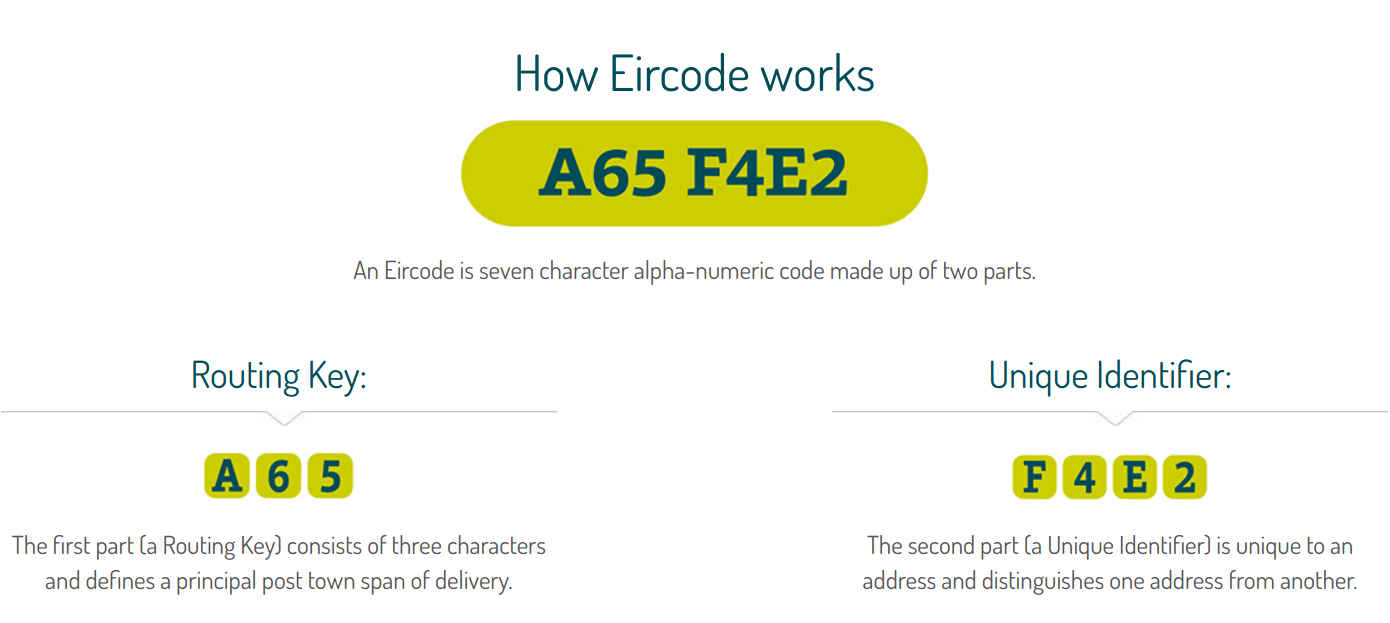
#### Write a function in Python that takes a string as input and uses regular expressions to:

#### 1.1 check if the given string is a valid Eircode or not

#### 1.2 if the string is a valid Eircode, identify and print out the Eircode's geographical district

[3 marks]

* Each [Eircode](http://www.eircode.ie/what-is-eircode) consists of seven CAPITAL letters and/or digits, in the format A65B2CD. The first three characters together represent the "Routing Key" part of the code and correspond to one of 139 unique geographical districts.
* You can download the full list of routing keys in CSV format from [here](https://gist.githubusercontent.com/ajoorabchi/eac194a79dd26de8864f9206b7842ff1/raw/8ea1d8d5f74b5b2724e378b43d4df6094990c7db/Eircode%2520Routing%2520Key%2520Boundaries.csv) using the *!wget* command.
* Use the ['re' library](https://www.w3schools.com/python/python_regex.asp) for regular expressions in Python
* [Regex101](https://regex101.com/) is a handy website for trying and testing your regular expressions

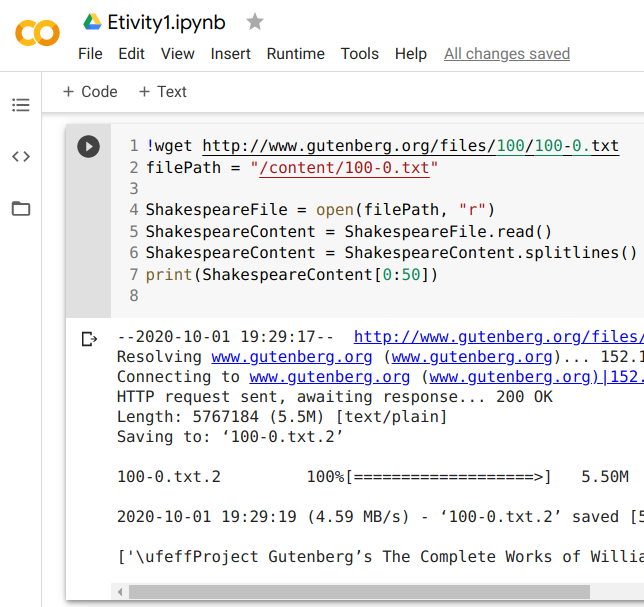


#### Expected Output:

#### 

### Task 2:

#### 2.1 Use the *!wget* command to download the Complete Works of William Shakespeare from [here](http://www.gutenberg.org/files/100/100-0.txt); then open the downloaded text file and print out its first 50 lines.

HINT: Python String [splitlines()](https://www.w3schools.com/python/ref_string_splitlines.asp) Method Split a string into a list where each line is a list item. 

#TASK 2.1

!wget http://www.gutenberg.org/files/100/100-0.txt

filePath = "/content/100-0.txt"

ShakespeareFile = open(filePath, "r")

ShakespeareContent = ShakespeareFile.read()

ShakespeareContent = ShakespeareContent.splitlines() # The splitlines() method splits a string into a list. The splitting is done at line breaks.

print(ShakespeareContent[0:50])

#### 

#### 

#### 

#### 

#### 2.2 Use the [tf.keras.preprocessing.text.Tokenizer](https://www.tensorflow.org/api_docs/python/tf/keras/preprocessing/text/Tokenizer) class to:

1. Tokenize the Shakespeare corpus
2. Print out the total number of Tokens in the corpus
3. print out the total number of Types in the corpus
4. Print out the most frequent Type in corpus and its frequency

[3 marks]

#### 2.3 Use the [PorterStemmer](https://www.nltk.org/api/nltk.stem.html#module-nltk.stem.porter) and [WordNetLemmatizer](https://www.nltk.org/api/nltk.stem.html#module-nltk.stem.wordnet) classes in the [NLTK.stem](https://www.nltk.org/api/nltk.stem.html) package to:

1. stem all the Types in the Shakespeare corpus; print out the total number of Types in the corpus after stemming.
2. Lemmatize all the Types in the Shakespeare corpus; print out the total number of Types in the corpus after lemmatization.
3. [Assert](https://www.w3schools.com/python/ref_keyword_assert.asp) the validity of this expression: *total\_number\_of\_types* > *total\_number\_of\_lemmatized\_types* > *total\_number\_of\_stemmed\_types*

*#TASK 2.3*

*from nltk.stem import PorterStemmer*

*import nltk*

*nltk.download('wordnet')*

*from nltk.stem.wordnet import WordNetLemmatizer*

*ps = PorterStemmer()*

*lemmatizer = WordNetLemmatizer()*

*YOUR CODE HERE*

[2 marks]

#### 2.4 Use the [Sentence Segmentation module in the spaCy package](https://spacy.io/usage/linguistic-features#sbd) to:

1. Segment the last 100 lines of the Shakespeare corpus into sentences
2. print out the segmented sentences and their total number.

HINT**:** 100 [lines in a text file](http://en.wikipedia.org/wiki/line_(text_file)) is just that, 100 lines; think of it as 100 lines on a sheet of paper; how many sentences are in that sheet? No way to know until you read and count. One sentence could occupy just half a line, whereas another sentence could occupy multiple lines. lines on a sheet of paper are physical concepts, whereas sentences are grammatical concepts. distinguishing the difference between the two is the goal of this exercise.

[2 marks]

#### </END OF ETIVITY>

Resources for Data Science and Statistical Learning students:

* [A gentle introduction to OOP in Python (classes, methods, attributes, objects)](https://www.youtube.com/watch?v=f0TrMH9s-VE)

Teaching team:

* Addition of BPE
* Level of difficulty (easy/fair/hard)
* TF tokenizer vs.NLTK tokenizer
* Hint to regex grouping
* [String Fundamentals, Concatenation, Indexing and Slicing: Python Basics](https://www.youtube.com/watch?v=WGHBaMOeamQ)
* Text to word cloud