```
1  import re
2  from importlib.resources.readers import remove_duplicates
3
4
5  class Data:
6
7      # list of possible type of questions that will be used in conditional statements [non-exhaustive]
8      possibleList = [["capital", "distance", "weather", "movie", "forecast"], # direct-answer question
9                      ["exercis", "diet", "cook"], # health-related questions
10                     ["calendar", "reminder"], # productivity questions
11                     ["score", "gam", "jok", "song"], # entertainment questions
12                     ["plu", "minus", "multipl", "divid", "formula", "concept"], # mathematical questions
13                     ["pric", "mean", "fact", "happen", "latest"], # knowledge-building question
14                     ["best", "advic", "help", "tip", "plan"]] # advice-seeking questions
15
16     # list of possible cities that the user might reference [non-exhaustive]
17     specificPlaceList = ["Los Angeles", "Chicago", "San Francisco", "Miami", "Austin", "Las Vegas", "
   Paris", "London", "Tokyo", "Sydney", "Rome", "Barcelona",
18     "Berlin", "Dubai", "Toronto", "Seoul", "Bangkok", "Mexico City", "Cape Town", "California", "
   Florida", "Texas", "New York", "Nevada", "Hawaii", "Colorado", "Alaska", "Arizona",
19     "Utah", "Illinois", "Michigan", "Washington", "Georgia", "North Carolina", "Tennessee", "South
   Carolina", "Oregon", "New Jersey", "Virginia", "United States", "Canada", "United Kingdom",
20     "France", "Italy", "Spain", "Mexico", "Germany", "Australia", "Brazil", "Japan", "India", "South
   Korea", "Thailand", "South Africa", "China", "Russia",  "Egypt", "Argentina", "New Zealand"]
21
22     # method that removes accidental duplicates found by regular expression
23     def __remove_duplicate(self, list):
24         accList = []
25         for i in list:
26             if i not in accList:
27                 accList.append(i)
28         return accList
29
30
31     # uses Python Regular Expressions to derive key data in a structural format, replicating a basic version of
   Natural Language Processing
32     # "QT" = Question Type && "I" = Identifier
33     def parsedData(self, userInput):
34         question_type = self.__remove_duplicate(re.findall(r"(what|who|why|where|when|how|will|can|play|
   should|is)", userInput))
35         identifiers = self.__remove_duplicate(re.findall(r"(capital|best|advic|help|tip|distance|plan|weather|
   forecast|latest|happen|movie|exercis|song|diet|calendar|reminder|cook|score|pric|mean|plu|minus|
   multipl|divid|jok|gam|fact|formula|concept)", userInput))
36
37         # returns error if either one of the variables above is empty, else, normal dictionary returned
38         if len(question_type) + len(identifiers) < 2:
39             issue = {"QT": "ERROR", "I": "ERROR"}
40             return issue
41         else:
42             result = {"QT": question_type, "I": identifiers}
43             return result
44
45     # takes the original sentence inputted by the user and then removes suffixes; local change not global
46     def stemWord(self, userInput):
47         return re.sub(r'\b(?!(is\b))(ing|ed|s)\b', '', userInput)
```