ASSIGNMENT NO:-5

import re

import long\_responses as long

def message\_probability(user\_message, recognised\_words, single\_response=False, required\_words=[]):

    message\_certainty = 0

    has\_required\_words = True

    # Counts how many words are present in each predefined message

    for word in user\_message:

        if word in recognised\_words:

            message\_certainty += 1

    # Calculates the percent of recognised words in a user message

    percentage = float(message\_certainty) / float(len(recognised\_words))

    # Checks that the required words are in the string

    for word in required\_words:

        if word not in user\_message:

            has\_required\_words = False

            break

    # Must either have the required words, or be a single response

    if has\_required\_words or single\_response:

        return int(percentage \* 100)

    else:

        return 0

def check\_all\_messages(message):

    highest\_prob\_list = {}

    # Simplifies response creation / adds it to the dict

    def response(bot\_response, list\_of\_words, single\_response=False, required\_words=[]):

        nonlocal highest\_prob\_list

        highest\_prob\_list[bot\_response] = message\_probability(message, list\_of\_words, single\_response, required\_words)

    # Responses -------------------------------------------------------------------------------------------------------

    response('Hello!', ['hello', 'hi', 'hey', 'sup', 'heyo'], single\_response=True)

    response('See you!', ['bye', 'goodbye'], single\_response=True)

    response('I\'m doing fine, and you?', ['how', 'are', 'you', 'doing'], required\_words=['how'])

    response('You\'re welcome!', ['thank', 'thanks'], single\_response=True)

    response('Thank you!', ['i', 'love', 'code', 'palace'], required\_words=['code', 'palace'])

    # Longer responses

    response(long.R\_ADVICE, ['give', 'advice'], required\_words=['advice'])

    response(long.R\_EATING, ['what', 'you', 'eat'], required\_words=['you', 'eat'])

    best\_match = max(highest\_prob\_list, key=highest\_prob\_list.get)

    # print(highest\_prob\_list)

    # print(f'Best match = {best\_match} | Score: {highest\_prob\_list[best\_match]}')

    return long.unknown() if highest\_prob\_list[best\_match] < 1 else best\_match

# Used to get the response

def get\_response(user\_input):

    split\_message = re.split(r'\s+|[,;?!.-]\s\*', user\_input.lower())

    response = check\_all\_messages(split\_message)

    return response

# Testing the response system

while True:

    print('Bot: ' + get\_response(input('You: ')))

import random

R\_EATING ="I don't like eating anything because i am bot obviously!"

R\_ADVICE = "If i were you, I would go to internet and type exactly what you wrote there!"

def unknown():

    response=["could you please re-phrase that? ",

              "...",

              "Sounds about right.",

              "what does  that mean?"][random.randrange(4)]

    return response

OUTPUT:-

PS C:\Users\DELL\OneDrive\Documents\K.AI> python -u "c:\Users\DELL\OneDrive\Documents\K.AI\main.py"

You: hi

Bot: Hello!

You: you are beutiful

Bot: Sounds about right.

You: