By Team#11(TR section): Mohammad Jaafari, Abdelrahman Elkadi and Nour Kallout

 $content = file.read().replace('\n', '')$ #transform file content to a string and remove empty lines for better processing

All imports:

In [2]: with open('covid.txt', 'r') as file:

```
In [1]: import random #using random in order to randomly pick replies for brainy from lists
        from sklearn.metrics.pairwise import cosine_similarity, cosine_distances #needed for calculating the cosine similarity
        from scipy.spatial import distance #needed for cosine similarity
        from sklearn.feature_extraction.text import CountVectorizer #needed for vectorization
        from sklearn.feature_extraction.text import TfidfVectorizer #alternative vectorization from bag of words
        warnings.filterwarnings('ignore') #needed to filter warnings (aesthetic purposes)
        from gtts import gTTS #gtts is needed for google text to speech in order to get brainy's voice
        from nltk import pos_tag, word_tokenize, sent_tokenize #needed for tokenization
        import os #needed in order to download the brainy voice mp3 files
        from playsound import playsound #needed in order to play the sound in notebook
        import time #needed for mp3 brainy unique file name
        import pandas as pd #pandas is needed for vectorization and cosine similarity
        import speech_recognition as sr #needed for speech recognition
        r = sr.Recognizer()
        import itertools #needed to turn a 4D list into a 1D list
        import re #regex is needed to clean the paragraph
```

Module #1: Load and Preprocess text data.

def preprocessing(document): #a simple preprocessing function for demonstration
 sentences = sent_tokenize(document) #we sentence tokenize the document

sentences = [word_tokenize(sent) for sent in sentences] #we word tokenize the sentences sentences = [pos_tag(sent) for sent in sentences] #we tag all words and return them

```
return(sentences)
             def turn_to_sentences(content): #needed to tidy and clean the paragraph
                    content = content.replace('â€'', ' ') #we replace unneeded characters with blank characters
                   content = re.sub(r'\[\d+\]', "", content) #we remove the wikipedia citation
                   return sent_tokenize(content) #we return the paragraph
              sentences = turn_to_sentences(content) #we turn the clean paragraph into sentences
             print(preprocessing(content)) #show pos tagging demonstration
           [[('Coronavirus', 'NNP'), ('disease', 'NN'), ('2019', 'CD'), ('(', '('), ('COVID-19', 'NNP'), ('is', 'VBZ'), ('a', 'DT'), ('contagious', 'JJ'), ('disease', 'NN'), ('caused', 'VBN'), ('by', 'IN'), ('a', 'DT'), ('virus', 'N
          N'), (',', ','), ('the', 'DT'), ('severe', 'JJ'), ('acute', 'JJ'), ('respiratory', 'NN'), ('syndrome', 'NN'), ('2', 'CD'), ('(', '('), ('SARS-CoV-2', 'NNP'), (')', ')'), ('.', '.')], [('The', 'DT'), ('first', 'J'), ('acute', 'JJ'), ('acute', 'JJ'), ('acute', 'JJ'), ('acute', 'NN'), ('syndrome', 'NN'), ('2', 'CD'), ('(', '('), ('SARS-CoV-2', 'NNP'), (')', ')'), ('.', '.')], [('The', 'DT'), ('first', 'J'), ('acute', 'JJ'), ('acute', 'JJ'), ('acute', 'JJ'), ('acute', 'NN'), ('acut
          J'), ('known', 'JJ'), ('case', 'NN'), ('was', 'VBD'), ('identified', 'VBN'), ('in', 'IN'), ('Wuhan', 'NNP'), (',', ','), ('China', 'NNP'), (',', ','), ('in', 'IN'), ('December', 'NNP'), ('2019', 'CD'), ('.', '.')], [('[', 'RB'), ('7',
           'CD'), (']', 'VBD'), ('The', 'DT'), ('disease', 'NN'), ('has', 'VBZ'), ('since', 'IN'), ('spread', 'VBN'), ('worldwide', 'NN'), (',', ','), ('leading', 'VBG'), ('to', 'TO'), ('an', 'DT'), ('ongoing', 'JJ'), ('pandemic', 'NN'), ('.',
           '.')], [('[', 'RB'), ('8', 'CD'), (']', 'NN'), ('Symptoms', 'NNP'), ('of', 'IN'), ('COVID-19', 'NNP'), ('are', 'VBP'), ('yariable', 'JJ'), (',', ','), ('but', 'CC'), ('often', 'RB'), ('include', 'VBP'), ('fever', 'RB'), (',', ','),
           ('[', 'VBD'), ('9', 'CD'), (']', 'JJ'), ('cough', 'NN'), (',', ','), ('headache', 'NN'), (',', ','), ('[', 'VBZ'), ('10', 'CD'), (']', 'NNP'), ('fatigue', 'NN'), (',', ','), ('breathing', 'VBG'), ('difficulties', 'NNS'), (',', ',')
           ('loss', 'NN'), ('of', 'IN'), ('smell', 'NN'), (',', ','), ('and', 'CC'), ('loss', 'NN'), ('.', '.')], [('[', 'RB'), ('11', 'CD'), (']', 'JJ'), ('[', '$'), ('12', 'CD'), (']', 'NNP'), ('[', 'VBD'), ('13', 'IN'), ('smell', 'NN'), ('smell', 'NN')
            'CD'), (']', 'NN'), ('Symptoms', 'NNP'), ('may', 'MD'), ('begin', 'VB'), ('one', 'CD'), ('to', 'TO'), ('days', 'NNS'), ('after', 'IN'), ('exposure', 'NN'), ('to', 'TO'), ('the', 'DT'), ('virus', 'NN'), ('.', '.')],
           [('At', 'IN'), ('least', 'JJS'), ('a', 'DT'), ('third', 'JJ'), ('of', 'IN'), ('people', 'NNS'), ('who', 'WP'), ('are', 'VBP'), ('do', 'VBP'), ('not', 'RB'), ('develop', 'VB'), ('noticeable', 'JJ'), ('symptoms', 'NN
          S'), ('.', '.')], [('[', 'RB'), ('14', 'CD'), (']', 'NN'), ('0f', 'IN'), ('those', 'DT'), ('people', 'NNS'), ('who', 'WP'), ('symptoms', 'NNS'), ('noticeable', 'JJ'), ('enough', 'RB'), ('to', 'TO'), ('be', 'VB'),
           '('), ('up', 'IN'), ('to', 'TO'), ('mild', 'VB'), ('pneumonia', 'NN'), (')', ')'), (',', ','), ('while', 'IN'), ('4', 'CD'), ('%', 'NN'), ('severe', 'JJ'), ('symptoms', 'NNS'), ('(', '('), ('dyspnea', 'NN'), (',',
           ','), ('hypoxia', 'NN'), (',', ','), ('or', 'CC'), ('more', 'JJR'), ('than', 'IN'), ('50', 'CD'), ('%', 'NN'), ('involvement', 'NN'), ('on', 'IN'), ('imaging', 'VBG'), (')', ')'), (',', ','), ('and', 'CC'), ('5', 'CD'),
           ('%', 'NN'), ('suffer', 'NN'), ('critical', 'JJ'), ('symptoms', 'NNS'), ('(', '('), ('respiratory', 'JJ'), ('failure', 'NN'), (',', ','), ('shock', 'NN'), (',', ','), ('or', 'CC'), ('multiorgan', 'JJ'), ('dysfunction', 'NN'), (')',
           ')'), ('.', '.')], [('[', 'RB'), ('15', 'CD'), (']', 'JJ'), ('Older', 'JJR'), ('people', 'NNS'), ('are', 'VBP'), ('at', 'IN'), ('higher', 'JJR'), ('risk', 'NN'), ('of', 'IN'), ('developing', 'VBG'), ('severe', 'JJ'), ('sym
           ptoms', 'NNS'), ('.', '.')], [('Some', 'DT'), ('people', 'NNS'), ('continue', 'VBP'), ('to', 'TO'), ('experience', 'VB'), ('a', 'DT'), ('of', 'IN'), ('effects', 'NNS'), ('(', '('), ('long', 'RB'), ('COVID', 'NNP'),
           (')', ')'), ('for', 'IN'), ('months', 'NNS'), ('after', 'IN'), ('recovery', 'NN'), (',', ','), ('and', 'CC'), ('damage', 'NN'), ('to', 'T0'), ('organs', 'NNS'), ('has', 'VBZ'), ('been', 'VBN'), ('observed', 'VBN'), ('.', '.')], [('[',
           'RB'), ('16', 'CD'), (']', 'JJ'), ('Multi-year', 'JJ'), ('studies', 'NNS'), ('are', 'VBP'), ('underway', 'JJ'), ('further', 'JJ'), ('investigate', 'VB'), ('the', 'DT'), ('long-term', 'JJ'), ('effects', 'NNS'), ('of', 'I
          N'), ('the', 'DT'), ('disease', 'NN'), ('.', '.')], [('[', 'RB'), ('16', 'CD'), (']', 'JJ'), ('COVID-19', 'NNP'), ('when', 'WRB'), ('people', 'NNS'), ('breathe', 'VBP'), ('in', 'IN'), ('air', 'NN'), ('contaminate
           d', 'VBN'), ('by', 'IN'), ('droplets', 'NNS'), ('and', 'CC'), ('small', 'JJ'), ('particles', 'NNS'), ('containing', 'VBG'), ('the', 'DT'), ('virus', 'NN'), ('.', '.')], [('The', 'DT'), ('risk', 'NN'), ('of',
           ('breathing', 'VBG'), ('these', 'DT'), ('in', 'IN'), ('is', 'VBZ'), ('highest', 'JJS'), ('when', 'WRB'), ('people', 'NNS'), ('are', 'VBP'), ('in', 'IN'), ('close', 'JJ'), ('proximity', 'NN'), (',', ','), ('but', 'CC'), ('they', 'PRP'),
           ('can', 'MD'), ('be', 'VB'), ('inhaled', 'VBN'), ('over', 'IN'), ('longer', 'JJR'), ('distances', 'NNS'), (',', ','), ('particularly', 'RB'), ('indoors', 'NNS'), ('.', '.')], [('Transmission', 'NN'), ('can', 'MD'), ('also', 'RB'), ('oc
           cur', 'VB'), ('if', 'IN'), ('splashed', 'VBN'), ('or', 'CC'), ('sprayed', 'VBN'), ('with', 'IN'), ('fluids', 'NNS'), ('in', 'IN'), ('the', 'DT'), ('eyes', 'NNS'), (',', ','), ('nose', 'RB'), ('or', 'CC'), ('mout
          h', 'VB'), (',', ','), ('and', 'CC'), (',', ','), ('rarely', 'RB'), (',', ','), ('via', 'IN'), ('surfaces', 'NNS'), ('.', '.')], [('People', 'NNS'), ('remain', 'VBP'), ('contagious', 'JJ'), ('for', 'IN'), ('u
          p', 'IN'), ('to', 'TO'), ('20', 'CD'), ('days', 'NNS'), (',', ','), ('and', 'CC'), ('can', 'MD'), ('spread', 'VB'), ('the', 'DT'), ('virus', 'NN'), ('even', 'RB'), ('if', 'IN'), ('they', 'PRP'), ('do', 'VBP'), ('not', 'RB'), ('develo
          p', 'VB'), ('symptoms', 'NNS'), ('.', '.')], [('[', 'RB'), ('17', 'CD'), (']', 'JJ'), ('18', 'CD'), (']', 'NNP'), ('several', 'JJ'), ('testing', 'VBG'), ('methods', 'NNS'), ('have', 'VBP'), ('been', 'VBN'), ('developed', 'V
           BN'), ('to', 'TO'), ('diagnose', 'VB'), ('the', 'DT'), ('disease', 'NN'), ('.', '.')], [('The', 'DT'), ('standard', 'JJ'), ('method', 'NN'), ('is', 'VBZ'), ('by', 'IN'), ('detection', 'NN'), ('of', 'IN'), ('the', 'DT'), ('the', 'DT'), ('diagnostic', 'JJ'), ('diagnostic', 'JJ'), ('method', 'NN'), ('is', 'VBZ'), ('by', 'IN'), ('diagnostic', 'NN'), ('of', 'IN'), ('the', 'DT'), ('diagnostic', 'JJ'), ('diagnostic', 'JJ'), ('diagnostic', 'NN'), ('diagnostic', 'NN'
           'DT'), ('virus', 'NN'), ("'s", 'POS'), ('nucleic', 'JJ'), ('acid', 'NN'), ('by', 'IN'), ('real-time', 'JJ'), ('reverse', 'NN'), ('polymerase', 'NN'), ('chain', 'NN'), ('reaction', 'NN'), ('(', '('), ('rRT-PCR',
           'JJ'), (')', ')'), (',', ','), ('transcription-mediated', 'JJ'), ('amplification', 'NN'), ('(', '('), ('TMA', 'NNP'), (')', ')'), (',', ','), ('or', 'CC'), ('by', 'IN'), ('reverse', 'JJ'), ('transcription', 'NN'), ('loop-mediated', 'J
          J'), ('isothermal', 'JJ'), ('amplification', 'NN'), ('(', '('), ('RT-LAMP', 'NNP'), (')', ')'), ('from', 'IN'), ('a', 'DT'), ('swab.Several', 'JJ'), ('COVID-19', 'NNP'), ('vaccines', 'NNS'), ('have', 'VBP'),
           ('been', 'VBN'), ('approved', 'VBN'), ('and', 'CC'), ('distributed', 'VBN'), ('in', 'IN'), ('various', 'JJ'), ('countries', 'NNS'), (',', ','), ('which', 'WDT'), ('have', 'VBP'), ('initiated', 'VBN'), ('mass', 'NN'), ('vaccination', 'N
          N'), ('campaigns', 'NNS'), ('.', '.')], [('Other', 'JJ'), ('preventive', 'JJ'), ('measures', 'NNS'), ('include', 'VBP'), ('physical', 'JJ'), ('or', 'CC'), ('social', 'JJ'), ('distancing', 'NN'), (',', ','), ('quarantining', 'VBG'),
           (',', ','), ('ventilation', 'NN'), ('of', 'IN'), ('indoor', 'JJ'), ('spaces', 'NNS'), (',', ','), ('covering', 'VBG'), ('coughs', 'NNS'), ('and', 'CC'), ('sneezes', 'NNS'), (',', ','), ('hand', 'NN'), ('washing', 'NN'), (',', ','), ('and', 'CC'), ('sneezes', 'NNS'), (',', ','), ('hand', 'NN'), ('washing', 'NN'), (',', ','), ('and', 'CC'), ('sneezes', 'NNS'), (',', ','), ('and', 'CC'), ('sneezes', 'NNS'), (',', ','), ('hand', 'NN'), ('washing', 'NN'), (',', ','), ('and', 'CC'), ('sneezes', 'NNS'), (',', ','), ('sneezes', 'NNS'), ('and', 'CC'), ('sneezes', 'NNS'), (
          nd', 'CC'), ('keeping', 'VBG'), ('unwashed', 'JJ'), ('hands', 'NNS'), ('away', 'RB'), ('from', 'IN'), ('face', 'NN'), ('.', '.')], [('The', 'DT'), ('use', 'NN'), ('of', 'IN'), ('face', 'NN'), ('masks', 'NNS'), ('or', 'C
          C'), ('coverings', 'NNS'), ('has', 'VBZ'), ('been', 'VBN'), ('recommended', 'VBN'), ('in', 'IN'), ('public', 'JJ'), ('settings', 'NNS'), ('to', 'TO'), ('minimize', 'VB'), ('the', 'DT'), ('risk', 'NN'), ('of', 'IN'), ('transmission', 'N
          N'), ('.', '.')], [('While', 'IN'), ('work', 'NN'), ('is', 'VBZ'), ('underway', 'JJ'), ('to', 'TO'), ('develop', 'VB'), ('that', 'WDT'), ('inhibit', 'VBP'), ('the', 'DT'), ('virus', 'NN'), (',', ','), ('the', 'DT'),
          ('primary', 'JJ'), ('treatment', 'NN'), ('is', 'VBZ'), ('symptomatic', 'JJ'), ('.', '.')], [('Management', 'NN'), ('the', 'DT'), ('treatment', 'NN'), ('of', 'IN'), ('symptoms', 'NNS'), (',', ','), ('supportive', 'J
          J'), ('care', 'NN'), (',', ','), ('isolation', 'NN'), (',', ','), ('and', 'CC'), ('experimental', 'JJ'), ('measures', 'NNS'), ('.', '.')]]
In [3]: sentences #show our sentences that are in a list to use for brainy
Out[3]: ['Coronavirus disease 2019 (COVID-19) is a contagious disease caused by a virus, the severe acute respiratory syndrome coronavirus 2 (SARS-COV-2).',
               'The first known case was identified in Wuhan, China, in December 2019.',
               'The disease has since spread worldwide, leading to an ongoing pandemic.Symptoms of COVID-19 are variable, but often include fever, cough, headache, fatigue, breathing difficulties, loss of smell, and loss of taste.',
               'Symptoms may begin one to fourteen days after exposure to the virus.',
               'At least a third of people who are infected do not develop noticeable symptoms.',
               'Of those people who develop symptoms noticeable enough to be classed as patients, most (81%) develop mild to moderate symptoms (up to mild pneumonia), while 14% develop severe symptoms (dyspnea, hypoxia, or more than 50% lung invol
              vement on imaging), and 5% suffer critical symptoms (respiratory failure, shock, or multiorgan dysfunction).',
               'Older people are at a higher risk of developing severe symptoms.',
               'Some people continue to experience a range of effects (long COVID) for months after recovery, and damage to organs has been observed.',
               'Multi-year studies are underway to further investigate the long-term effects of the disease.COVID-19 transmits when people breathe in air contaminated by droplets and small airborne particles containing the virus.',
               'The risk of breathing these in is highest when people are in close proximity, but they can be inhaled over longer distances, particularly indoors.',
```

Module #2: Greetings. & Module #6: Goodbyes.

Kindly note that you will be able to test everything when you run the last cell in our notebook.

'While work is underway to develop drugs that inhibit the virus, the primary treatment is symptomatic.', 'Management involves the treatment of symptoms, supportive care, isolation, and experimental measures.']

'The use of face masks or coverings has been recommended in public settings to minimize the risk of transmission.',

'Transmission can also occur if splashed or sprayed with contaminated fluids in the eyes, nose or mouth, and, rarely, via contaminated surfaces.',

'People remain contagious for up to 20 days, and can spread the virus even if they do not develop symptoms. Several testing methods have been developed to diagnose the disease.',

d isothermal amplification (RT-LAMP) from a nasopharyngeal swab. Several COVID-19 vaccines have been approved and distributed in various countries, which have initiated mass vaccination campaigns.",

'Other preventive measures include physical or social distancing, quarantining, ventilation of indoor spaces, covering coughs and sneezes, hand washing, and keeping unwashed hands away from the face.',

```
In [4]: greetings = ['hi', 'hello', 'good morning', 'morning', 'howdy', 'brainy'] #greetings for brainy to recognize

goodbyes = ['bye', 'good bye', 'goodbye', 'bye brainy', 'bye-bye', 'bye bye'] #goodbyes for brainy to recognize

greetings_response = ['Hello human!', 'Hello!', 'Good morning!'] #greetings to pick from randomly as replies

goodbye_response = ['Goodbye human!', 'Bye human!'] #goodbyes to pick from randomly as replies

speak = ['i need to speak', "i will speak", "record for me", "get my input", "speak"] #signals for brainy to get voice input from the user

confused = 'I am sorry, I do not understand what you are asking me.' #confused reply
```

"The standard diagnostic method is by detection of the virus's nucleic acid by real-time reverse transcription polymerase chain reaction (rRT-PCR), transcription-mediated amplification (TMA), or by reverse transcription loop-mediate

Module #4: Speech Recognition. & Module #5:

```
In [5]: def play(response): #function to play the sound of a response of brainy
            tts = gTTS(text = response, lang = 'en',tld="com") #we get the tts of the response off google translate
            date_string = str(time.time())
            file1 = str("nlp"+ date_string + ".mp3") #we name it a unique name using current time in order to avoid errors
            tts.save(file1)
            playsound(file1)
            os.remove(file1) #we save the sound file to play it, we play it, then we remove it
        def listen(): #function to listen for a response
            with sr.Microphone() as source: #we use our microphone as a source
                print("Brainy says: Talk") #brainy signals for us to talk
                audio_text = r.listen(source) #we listen and get an audio text
                print("Brainy says: Time is over, thanks.") #brainy signals for us to stop talking
                play("Time is over, thanks.")
                try: #we check if we recognize the audio and return it if we do, else brainy signals for us that he did not
                    print("Text: "+r.recognize_google(audio_text))
                    return r.recognize_google(audio_text)
                except:
                    print("Brainy says: Sorry, I did not get that.")
                    play("Sorry, I did not get that.")
```

Module #3: Text Vectorization and Similarity Measure.

return "Sorry, I did not get that."

```
In [6]: def similar(input): #find the most similar question to the unser input
            cosines = [0,0,0,0,0]#a list containing cosine similarities with each of 5 questions
            questions = ["What is the coronavirus?","What are the symptoms of the coronavirus?","How does the coronavirus transmits?"
             ,"What are the testing methods for the coronavirus?","What are the preventative methods for the coronavirus?"]
        #a list containing covid questions to get cosine similarities from
            for i in range(5): #we check with all 5 questions
                bow_list = [input, questions[i]] #we make a bag of words list
                count_vector = CountVectorizer(ngram_range = (1, 1)) #we make a count vectorizer
                bow = count_vector.fit_transform(bow_list) #we transform the bag of words list into a bag of words
                bow_features = count_vector.get_feature_names() #we get the feature names
                df = pd.DataFrame.sparse.from_spmatrix(bow, index=bow_list, columns=bow_features) #we create a dataframe for our bow
                cosines[i] = cosine_similarity([list(df.iloc[0])], [list(df.iloc[1])]) #we get the consine similarities form
                #the first and second rows in the dataframe
            cosines = list(itertools.chain(*list(itertools.chain(*cosines)))) #since our cosines in the cosine list are in 4D
            #we turn them into 1D elements of a list
            if max(cosines) < 0.5: #we have 0.5 as the threshold of our similarity, if all cosines are under 0.5 we return that
                return confused #brainy is confused
            index = cosines.index(max(cosines)) #we get the index of the maximum cosine similarity
            if index == 0: #if its the first question that is most similar we return the reply for it which is the first sentence in
                return sentences[0] #the paragraph
            elif index == 1: #similar to above but its the 3rd sentence for the 2nd question
                return sentences[3]
            elif index == 2: #similar to above but its 3 sentence replies
                return sentences[10] + " " +sentences[11] + " "+sentences[12]
            elif index == 3: #similar to above but its 2 sentences reply
                return sentences[14] + " " +sentences[15]
            else: #else it will be the final 5th question, similar concept to above
                return sentences[16] + " " +sentences[17]
```

Talk with out Conversational Agent, Brainy:

Please be patient with Brainy. It might take it some time to read your text/listen to your input and reply with text and through the speakers.

Kindly note that in order to talk to Brainy through the microphone you need to send 'speak' as input and wait for Brainy to ask you to talk.

```
In [7]: while True: #while loop for brainy
           user = input() #we get the user input in the box
            user = user.lower() #turn it to all lower so we can compare it easily with the lists
            if user in greetings: #behavior if greeted, replies with a greeting through chat and audio
                response = random.choice(greetings_response)
                print("Brainy says: "+ response)
                play(response)
            elif user in speak: #behavior if signaled to listen to the user speaking
                user = listen() #we run the listen command
                if(user == "Sorry, I did not get that."):
                    continue
                                                            #if we didnt get a good response we take care of that in
                                                            #the listen function so we just continue over the loop
                elif user in greetings: #we check if its a greeting spoken from the user
                    response = random.choice(greetings_response)
                    print("Brainy says: "+ response)
                    play(response)
                    continue
                elif user in goodbyes: #we check if its a goodbye spoken from the user
                    response = random.choice(goodbye_response)
                    print("Brainy says: "+ response)
                    play(response)
                    break #we break from the for loop
                    response = similar(user) #behavior if spoken to a question normally
                    print("Brainy says: "+ response)
                    play(response)
            elif user in goodbyes: #behavior if farewelled, replies with a greeting through chat and audio
                response = random.choice(goodbye_response)
                print("Brainy says: "+ response)
                play(response)
```

```
break #we break from the for loop
     else:
         response = similar(user) #behavior if typed to a question normally
         print("Brainy says: "+ response)
         play(response)
hi
Brainy says: Good morning!
what is the coronavirus?
Brainy says: Coronavirus disease 2019 (COVID-19) is a contagious disease caused by a virus, the severe acute respiratory syndrome coronavirus 2 (SARS-COV-2).
speak
Brainy says: Talk
Brainy says: Time is over, thanks.
Text: hi
Brainy says: Good morning!
speak
Brainy says: Talk
Brainy says: Time is over, thanks.
Text: hello
Brainy says: Hello!
speak
Brainy says: Talk
Brainy says: Time is over, thanks.
Text: what is the coronavirus
Brainy says: Coronavirus disease 2019 (COVID-19) is a contagious disease caused by a virus, the severe acute respiratory syndrome coronavirus 2 (SARS-COV-2).
speak
Brainy says: Talk
Brainy says: Time is over, thanks.
Text: bye
Brainy says: Goodbye human!
```