Report on

Joy of Programming using Python

Submitted for Summer Internship Program

By

Vishvajeet Singh(CSE,1910069)

Under the Guidance of -:

Dr. Pratima Singh

Mr.Binayak Parashar

AJAY KUMAR GARG ENGINEERING COLLEGE,GHAZIABAD

SESSION 2020-21

Speech-Text Recognition

IDE Used-:

Pycharm

Date-:06.07.2020

Group Members-:

Vishvajeet Singh -:	1900270100193
Vivek Kumar Gupta –:	1900270100194
Vikas Srivastava -:	1900270100187
Yash Sahu -:	1900270100197

CONTENTS-:

ABSTRACT-:

Suppose we are building a model and instead of a written approach we want our system to respond to speech, it becomes fairly difficult and requires a lot of data to be processed.

A speech recognition system overcomes this barrier by translating speech to text. So question arises…

- What is Speech Recognition?
- How Speech Recognition works ?
- What are the challenges faced?
- What are packages available for speech recognition in Python?

What is Speech Recognition?

Speech recognition is the ability of a machine or program to identify words and phrases in spoken language and convert them to a machine-readable format.

But speech recognition software has a limited vocabulary of words and phrases, and it may only identify these if they are spoken very clearly.

How Speech Recognition works?

Speech recognition system basically translates the spoken utterances to text. There are various real life examples of speech recognition system. For example- siri, google, alexa which takes the speech as input and translates it into text.

The advantage of using a speech recognition system is that it overcomes the barrier of literacy. A speech recognition model can serve both literate and illiterate audience as well, since it focuses on spoken utterances.

Challenges faced by Speech Recognition System-:

1.) Style of Speaking-:

Every individual person has a varied style of speaking, including accents as well.So,Pronunciation makes it difficult for a speech recognition system to translate the speech altogether.

2.) Environment-:

Environment adds a lot of background noise to the system as well. Even echo can add a lot of noise in the system as well.

3.) Speaker Characteristics-:

An old person's voice may not the be the same as that of an infant. The characteristics of a person's speech depends on many factors including the harshness and clarity as well.

4.) Language Constraints-:

Some spoken utterances may not have a viable meaning when it comes to translation.

After overcoming these challenges, it is fairly achievable for any speech recognition system to translate speech to text.

Packages available for speech recognition in Python-:

1.) Apiai
2.) Speech Recognition
3.) Google speech cloud
4.) assemblyai
5.) Pocketsphinx
6.) Watson developercloud
7.) Wit

INPUT FORMAT-:

How to install SpeechRecognition in Python?

To install SpeechRecognition package is python, run the following command in the terminal and it will be installed on your system.

```
$ pip install SpeechRecognition
```

Another approach to this, can be adding the package from the project interpreter if you are using Pycharm.

The package has a Recognizer class which is basically where the magic happens. It is basically a class which is used to recognize the speech. Following are seven methods which can read various audio sources using different APIs.

- recognize_bing( )
- recognize_google( )
- recognize_google_cloud( )
- recognize_houndify( )
- recognize_ibm( )
- recognize_wit( )
- recognize_sphinx( )

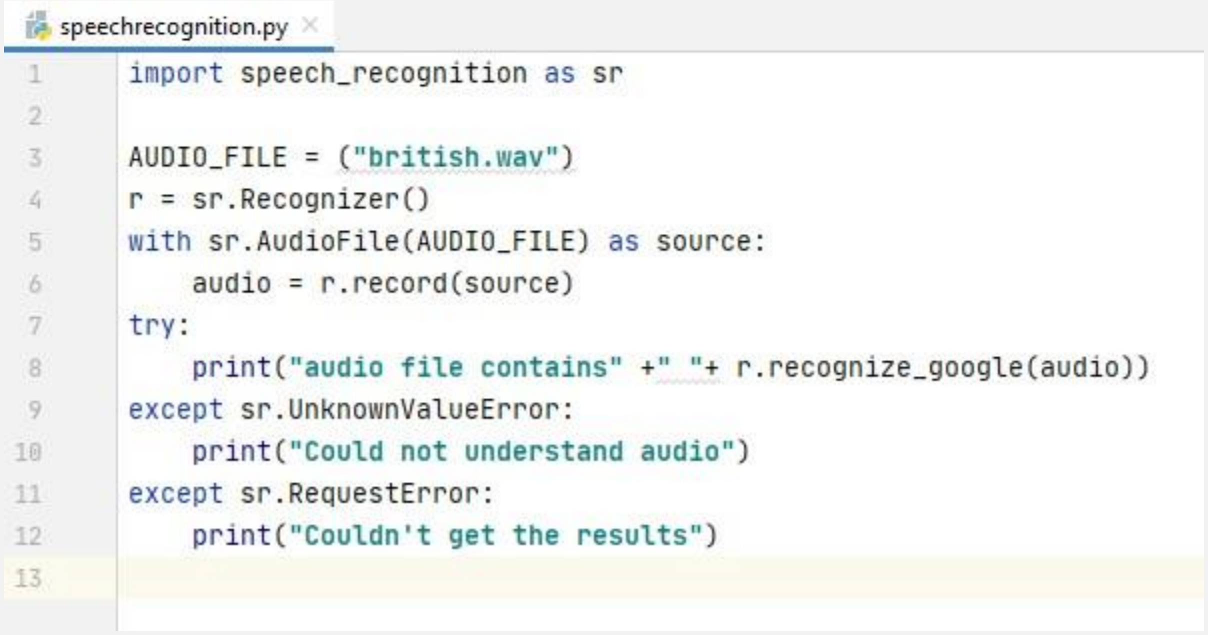
Taking Input From Audio Files(.wav file format)-:

SpeechRecognition makes working with audio files easy thanks to its handy AudioFile class. This class can be initialized with the path to an audio file and provides a context manager interface for reading and working with the file's contents.

Supported File Type-:

- WAV: must be in PCM/LPCM format
- AIFF
- AIFF-C
- FLAC: must be native FLAC format; OGG-FLAC is not supported

Using record() to capture a File-:



speechrecognition.py

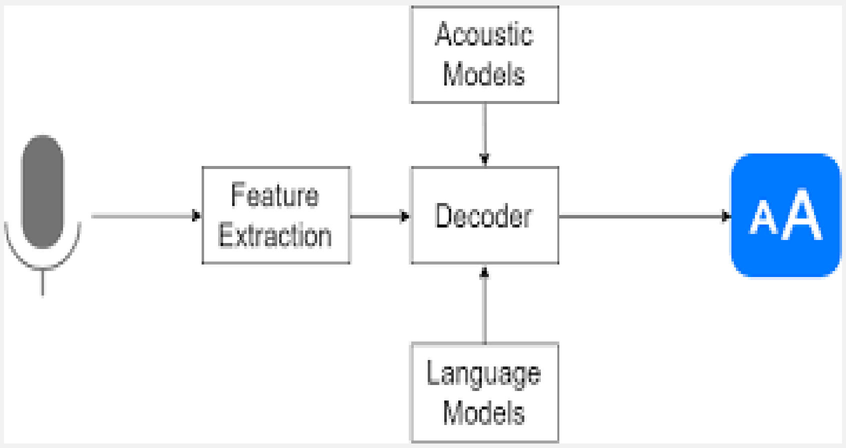
```
import speech_recognition as sr

AUDIO_FILE = ("british.wav")
r = sr.Recognizer()
with sr.AudioFile(AUDIO_FILE) as source:
    audio = r.record(source)
try:
    print("audio file contains" +" "+ r.recognize_google(audio))
except sr.UnknownValueError:
    print("Could not understand audio")
except sr.RequestError:
    print("Couldn't get the results")

```

The context manager opens the file and reads its contents, storing the data in an AudioFile instance called source.

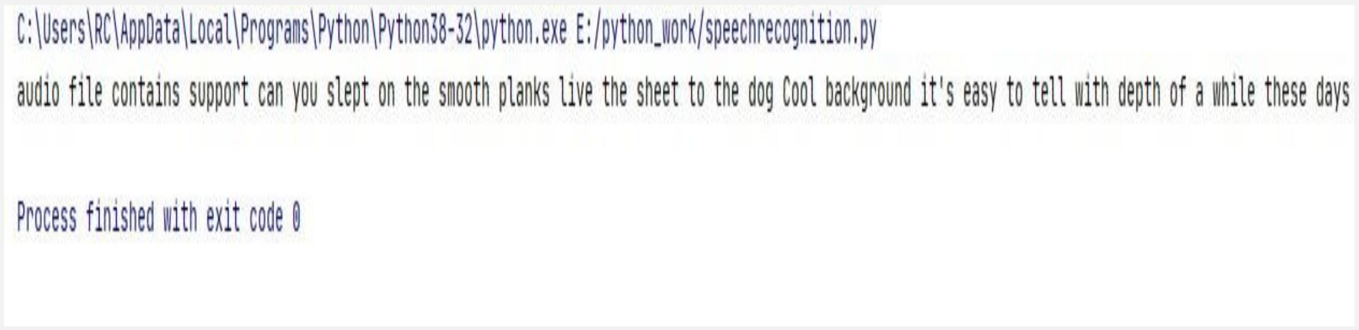
Then the record() method records the data from the entire file into an AudioData instance.

You can now invoke recognize_google() to attempt to recognize any speech in the audio.

Congratulations! You've just transcribed your first audio file!



OUTPUT FORMAT-:



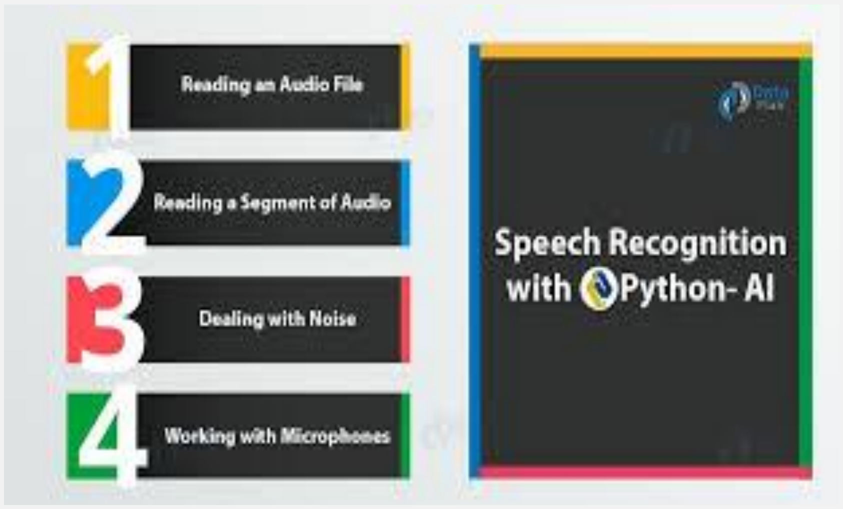
```
C:\Users\RC\AppData\Local\Programs\Python\Python38-32\python.exe E:/python_work/speechrecognition.py
audio file contains support can you slept on the smooth planks live the sheet to the dog Cool background it's easy to tell with depth of a while these days

Process finished with exit code 0
```

Speech recognition engine decodes the audio file and gives output as text of the words used in the audio file.

The above example is of one such audio file named ‘british.wav’ which is an english audio sample downloaded from harvard audio directories to just check whether the above written program is working properly or not.

As seen in the above snip, output is provided by printing :

“audio file contains + text of audio in british.wav”.



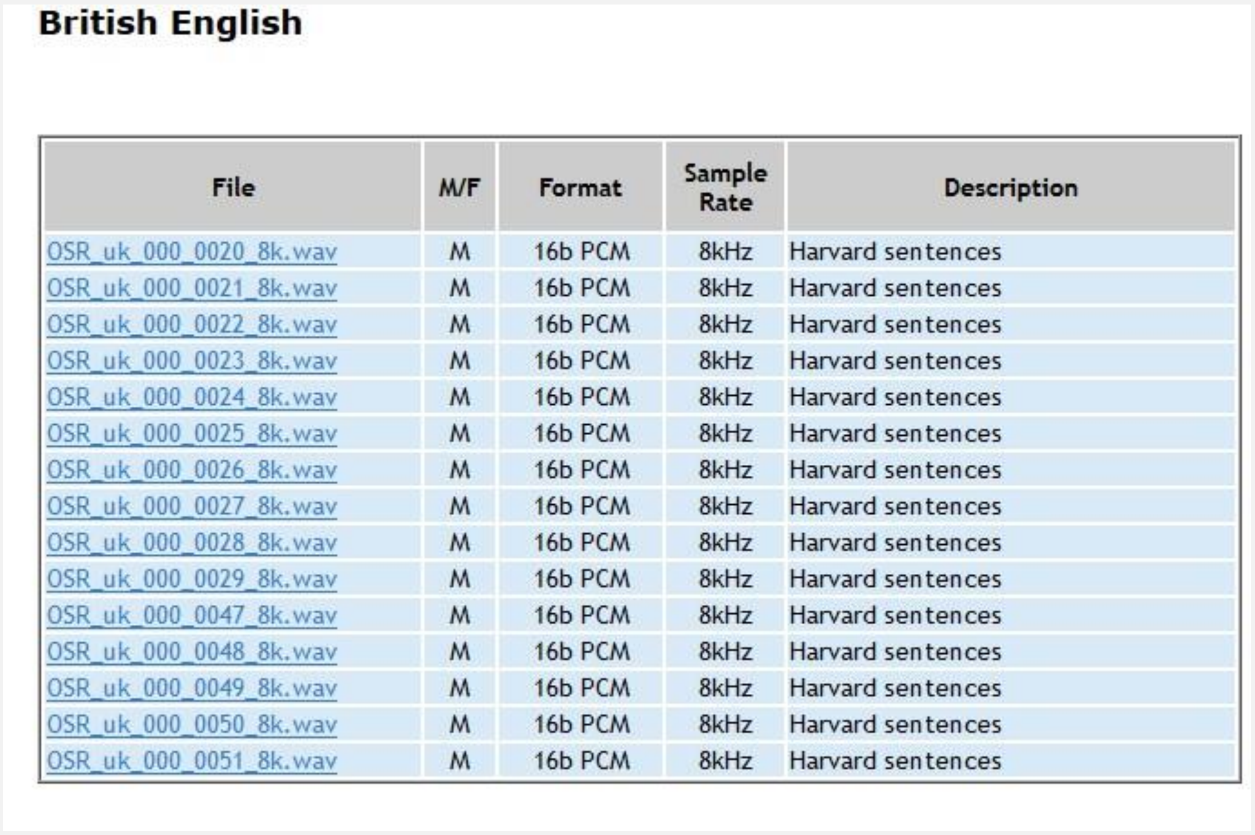
EXPLANATION-:

- first of all we will import speech_recognition as sr.
- Notice that we have speech_recognition in such format whereas earlier we have installed it in this way Spe,echRecognition , so you need to have a look around the cases because this is case sensitive.
- Now we have used as notation because writing speech_recognition whole every time is not a good way.
- Now we have to initialize r = sr.Recognizer() ,this will work as a recognizer to recognize our voice.
- So, with sr.AudioFile(AUDIO_FILE) as source: which means that we are initialising our source to Audio File .
- Next we will print a simple statement that recommend the user to speak anything.
- Now we have to use r.record(source) command and we have to listen the source.So, it will listen to the source and store it in the audio.
- It may happen some time the audio is not clear and you might not get it correctly ,so we can put it inside the try and except block .

- So inside the try block, our text will be text = r.recognize_google(audio).
- And this will convert our audio into text.
- Now we just have to print ("audio file contains" + r.recognize_google(audio)) ,this will print whatever you have said.
- In the except block we can just write (" Google Speech Recognition could not understand audio"),this will message you if your voice is not recorded clearly.

EXAMPLES(Sample Test Cases)

For example, we are going to take some sample audio files from Harvard open audio repository, and will check what are the results?

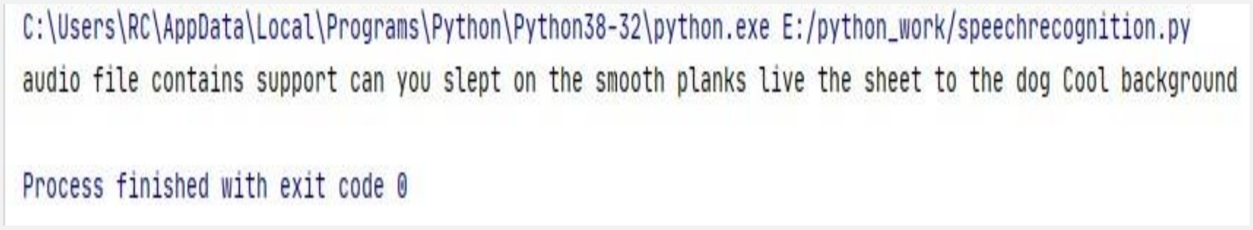


British English

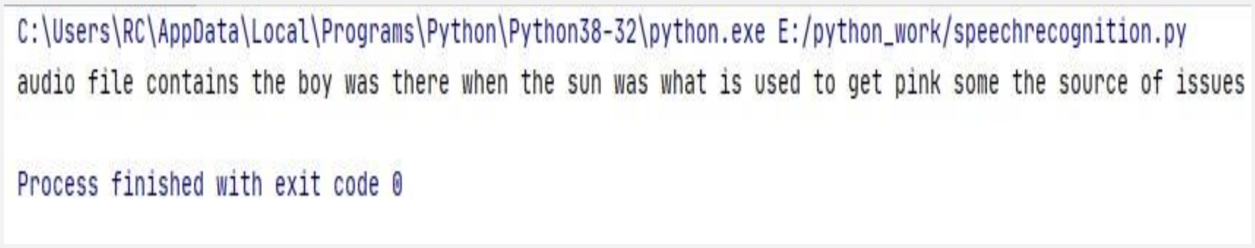
File	M/F	Format	Sample Rate	Description
OSR_uk_000_0020_8k.wav	M	16b PCM	8kHz	Harvard sentences
OSR_uk_000_0021_8k.wav	M	16b PCM	8kHz	Harvard sentences
OSR_uk_000_0022_8k.wav	M	16b PCM	8kHz	Harvard sentences
OSR_uk_000_0023_8k.wav	M	16b PCM	8kHz	Harvard sentences
OSR_uk_000_0024_8k.wav	M	16b PCM	8kHz	Harvard sentences
OSR_uk_000_0025_8k.wav	M	16b PCM	8kHz	Harvard sentences
OSR_uk_000_0026_8k.wav	M	16b PCM	8kHz	Harvard sentences
OSR_uk_000_0027_8k.wav	M	16b PCM	8kHz	Harvard sentences
OSR_uk_000_0028_8k.wav	M	16b PCM	8kHz	Harvard sentences
OSR_uk_000_0029_8k.wav	M	16b PCM	8kHz	Harvard sentences
OSR_uk_000_0047_8k.wav	M	16b PCM	8kHz	Harvard sentences
OSR_uk_000_0048_8k.wav	M	16b PCM	8kHz	Harvard sentences
OSR_uk_000_0049_8k.wav	M	16b PCM	8kHz	Harvard sentences
OSR_uk_000_0050_8k.wav	M	16b PCM	8kHz	Harvard sentences
OSR_uk_000_0051_8k.wav	M	16b PCM	8kHz	Harvard sentences

The above files are used in the speechrecognition.py in Pycharm and are decoded . Above shown are British English Samples of Harvard open audio repository

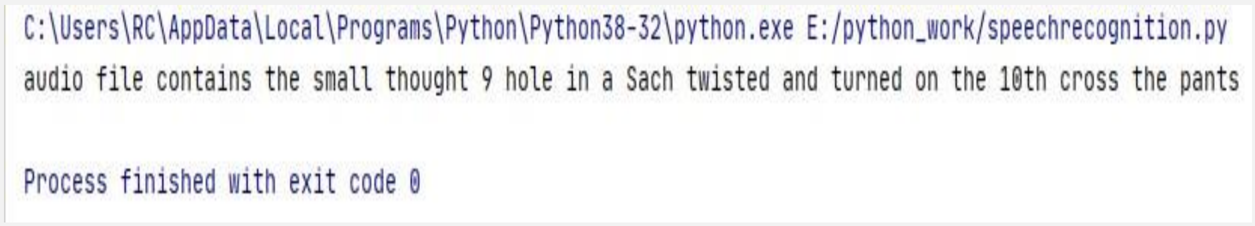
Results-:



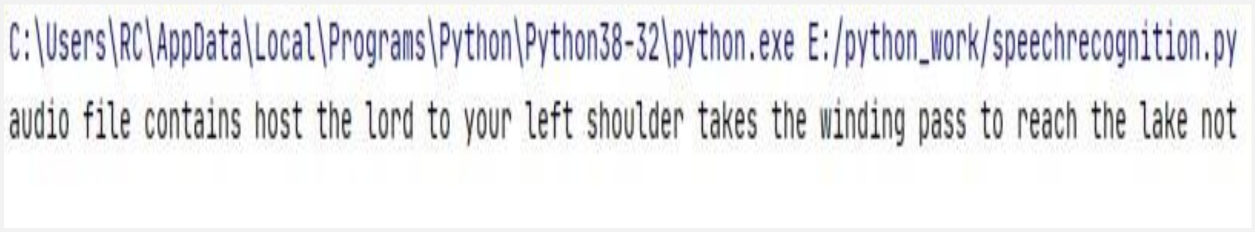
```
C:\Users\RC\AppData\Local\Programs\Python\Python38-32\python.exe E:/python_work/speechrecognition.py
audio file contains support can you slept on the smooth planks live the sheet to the dog Cool background

Process finished with exit code 0
```



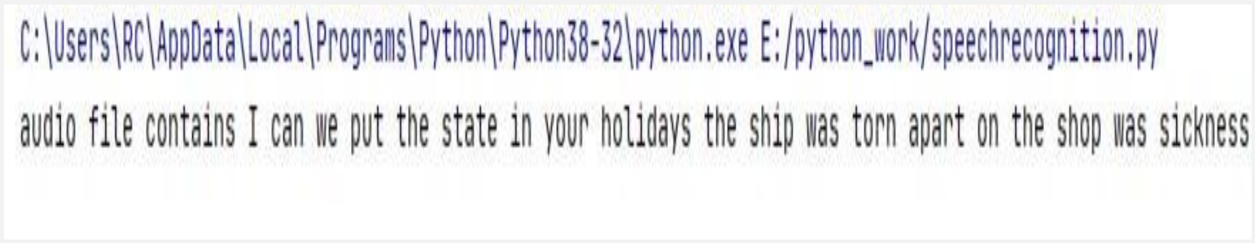
```
C:\Users\RC\AppData\Local\Programs\Python\Python38-32\python.exe E:/python_work/speechrecognition.py
audio file contains the boy was there when the sun was what is used to get pink some the source of issues

Process finished with exit code 0
```



```
C:\Users\RC\AppData\Local\Programs\Python\Python38-32\python.exe E:/python_work/speechrecognition.py
audio file contains the small thought 9 hole in a Sach twisted and turned on the 10th cross the pants

Process finished with exit code 0
```



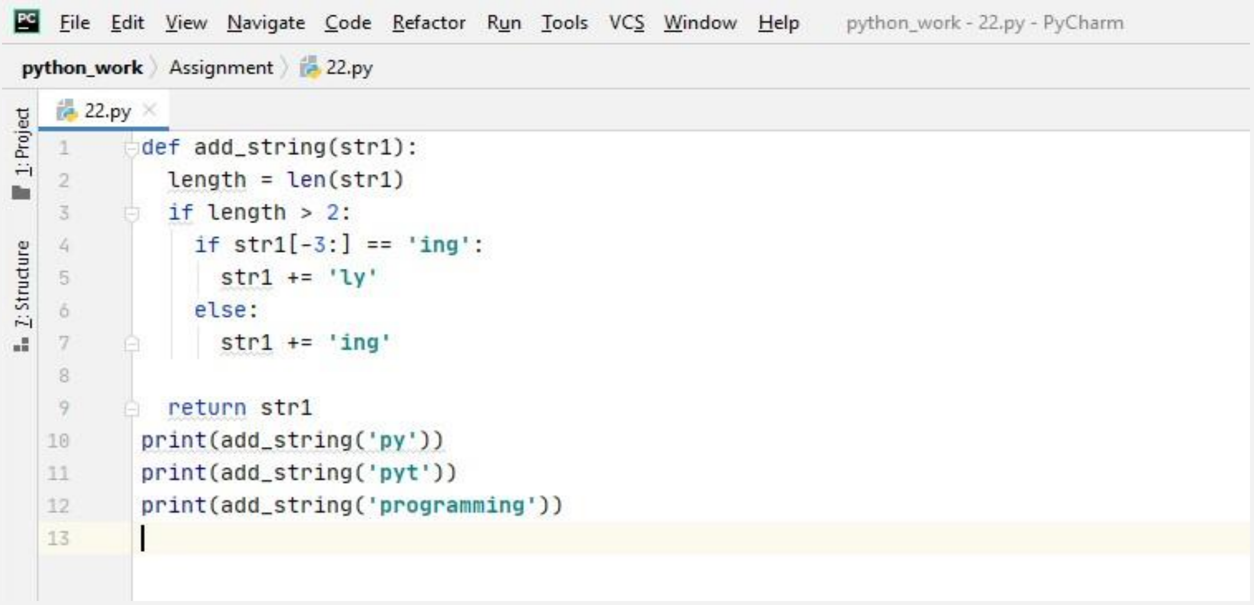
```
C:\Users\RC\AppData\Local\Programs\Python\Python38-32\python.exe E:/python_work/speechrecognition.py
audio file contains host the lord to your left shoulder takes the winding pass to reach the lake not
```



```
C:\Users\RC\AppData\Local\Programs\Python\Python38-32\python.exe E:/python_work/speechrecognition.py
audio file contains I can we put the state in your holidays the ship was torn apart on the shop was sickness
```

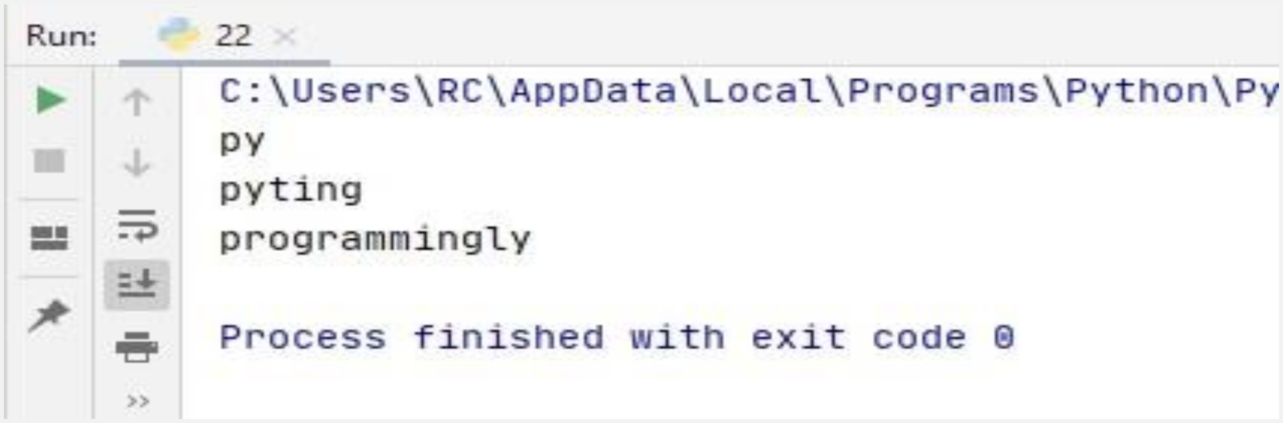
Assignments-:

1.)Write a Python program to add 'ing' at the end of a given string (length should be at least 3). If the given string already ends with 'ing' then add 'ly' instead. If the string length of the given string is less than 3, leave it unchanged


File Edit View Navigate Code Refactor Run Tools VCS Window Help python_work - 22.py - PyCharm
python_work Assignment 22.py
22.py
1: Project
Z: Structure

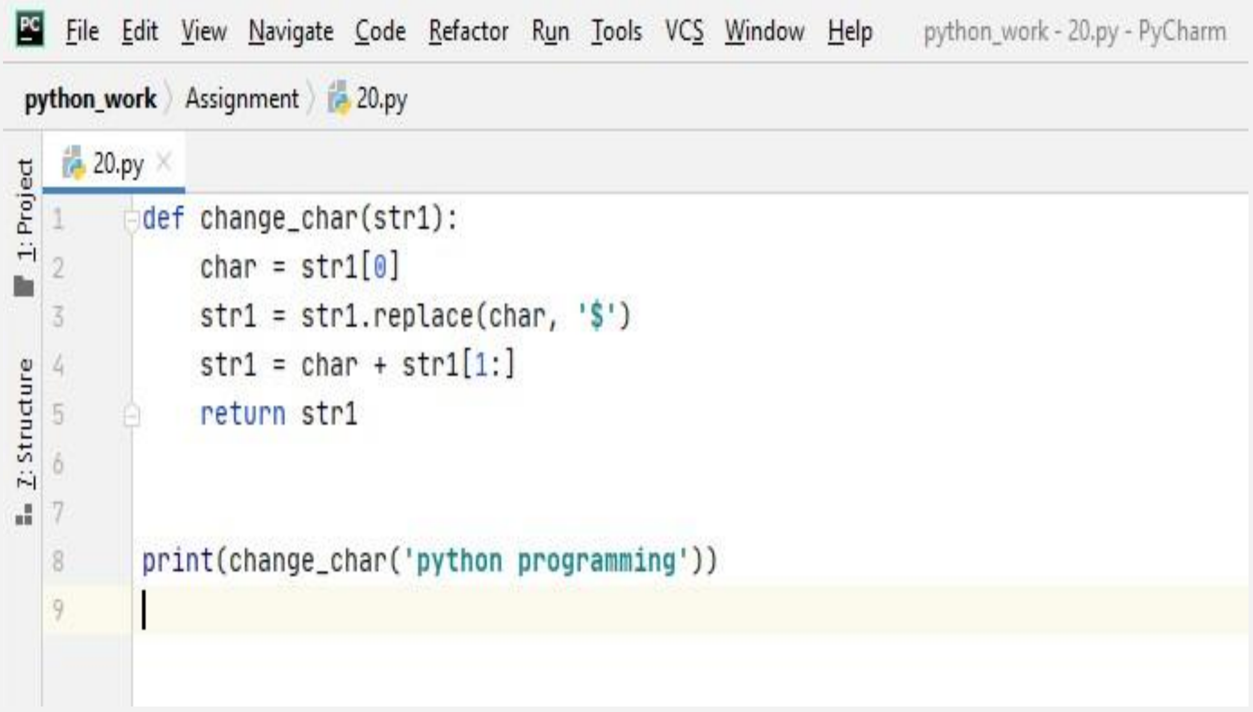
```python
def add_string(str1):
  length = len(str1)
  if length > 2:
    if str1[-3:] == 'ing':
      str1 += 'ly'
    else:
      str1 += 'ing'

  return str1
print(add_string('py'))
print(add_string('pyt'))
print(add_string('programming'))
```

Output-:



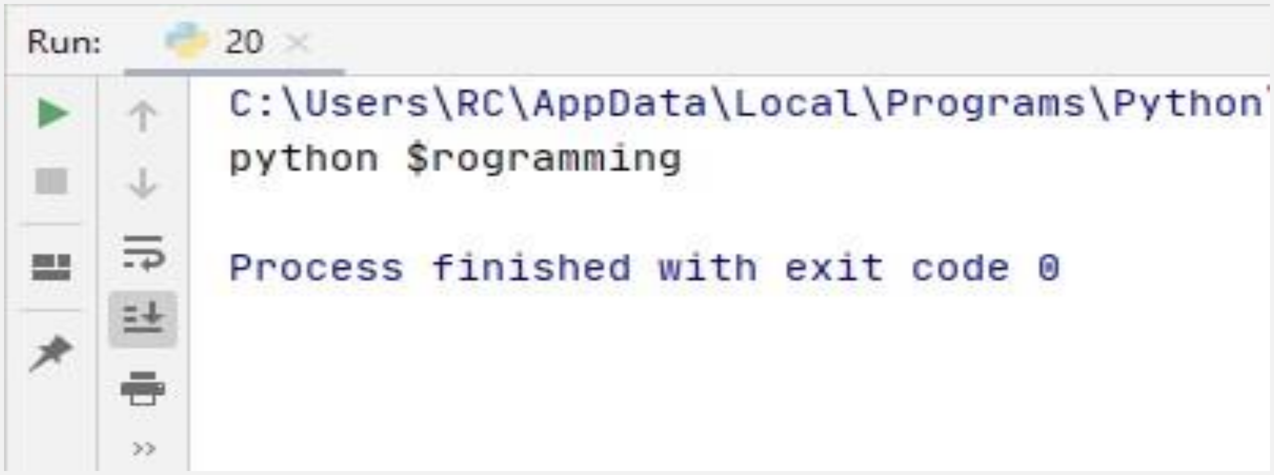
```
C:\Users\RC\AppData\Local\Programs\Python\Py
py
pyting
programmingly

Process finished with exit code 0
```

2.) Write a Python program to get a string from a given string where all occurrences of its firstchar have been changed to ‘$’ except the first char itself.


File Edit View Navigate Code Refactor Run Tools VCS Window Help python_work - 20.py - PyCharm
python_work 〉 Assignment 〉 20.py
20.py
1: Project
7: Structure

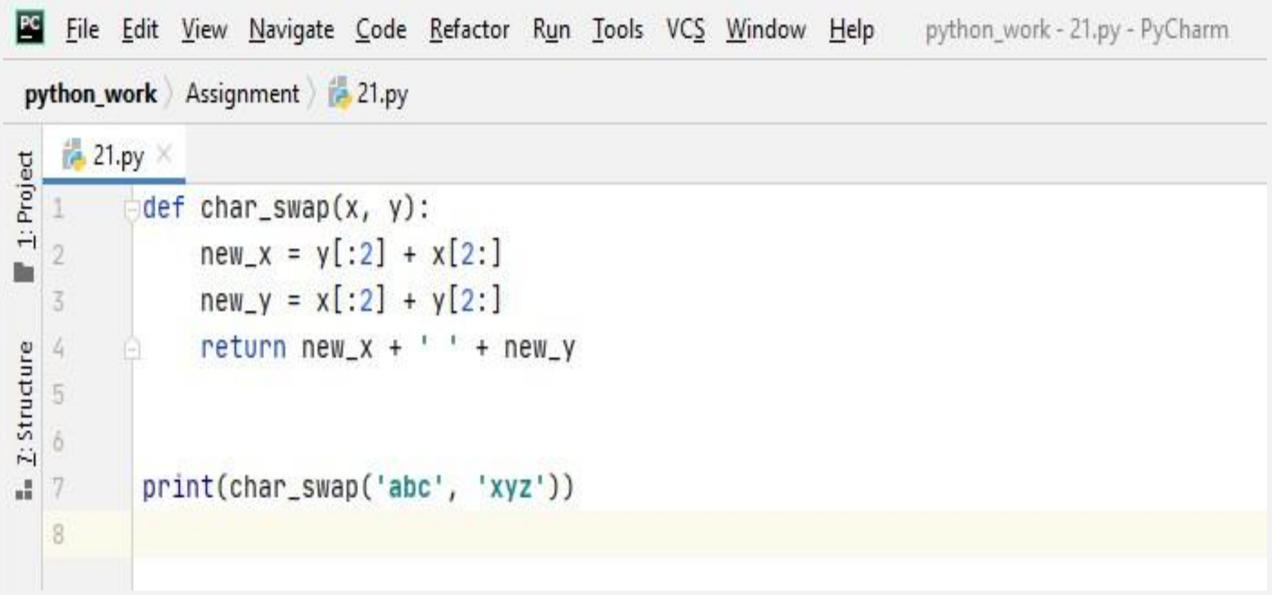
```
def change_char(str1):
    char = str1[0]
    str1 = str1.replace(char, '$')
    str1 = char + str1[1:]
    return str1


print(change_char('python programming'))
```

Output-:



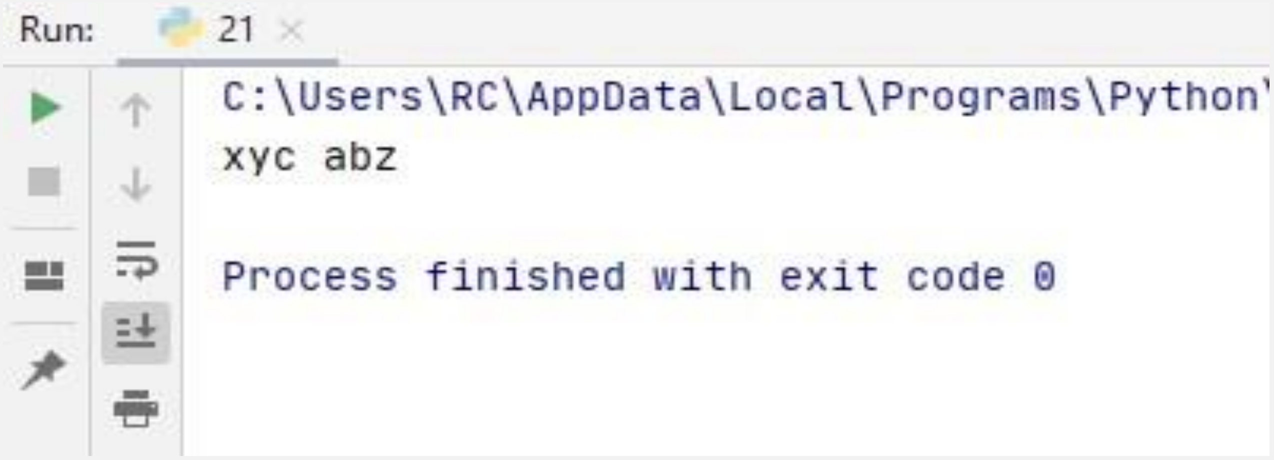
```
C:\Users\RC\AppData\Local\Programs\Python
python $rogramming

Process finished with exit code 0
```

3.) Write a Python program to get a single string from two given strings, separated by a space andswap the first two characters of each string.


File Edit View Navigate Code Refactor Run Tools VCS Window Help python_work - 21.py - PyCharm
python_work Assignment 21.py
21.py
1: Project
7: Structure

```
def char_swap(x, y):
    new_x = y[:2] + x[2:]
    new_y = x[:2] + y[2:]
    return new_x + ' ' + new_y


print(char_swap('abc', 'xyz'))
```

Output-:



```
C:\Users\RC\AppData\Local\Programs\Python
xyc abz

Process finished with exit code 0
```