Department of Electrical Engineering

Assignment

Date: 07/05/2020

Course title : *Programming Fundamentals*

Instructor : *Engr. M. Waqas*

Student Name: Saheeb ul Hassan

Student ID: *16318*

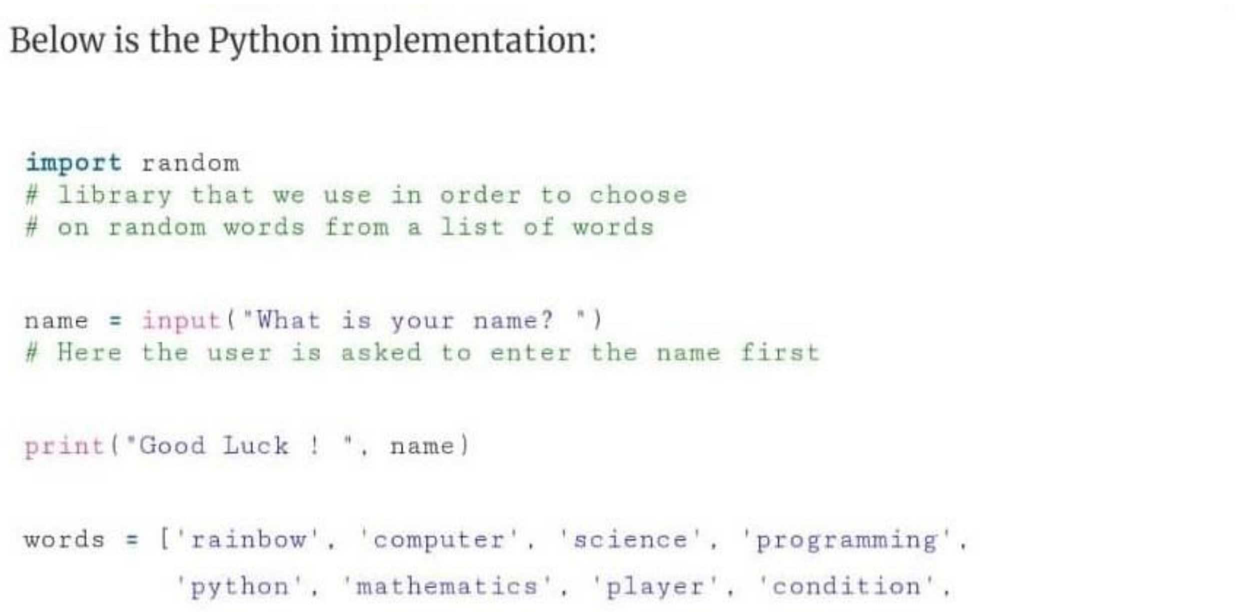
***QUESTION 1*:** Write a Guess the Word program in Python, The user needs to be able to input letter guesses. A limit should also be set on how many guesses they can use. This means you'll need a way to grab a word to use for guessing, this can be grabbed from a pre-made list. You will also need functions to check if the user has actually inputted a single letter, to check if the inputted letter is in the hidden word (and if it is, how many times it appears), to print letters, and a counter variable to limit guesses.

***QUESTION 2*:** Write a Password Generator program in Python, which generates a random password for the user. Ask the user how long they want their password to be (minimum 8 to 15 characters), how many letters, symbols and numbers they want in their password. Password generated MUST have a mix of upper and lowercase letters, as well as numbers and symbols.

***QUESTION 3*:** Write a Message Encryption Decryption program in Python, The user will input any text and your program must encrypt the text by using Base64 or HEX. The text must then be decrypted from the encrypted form to show that the decrypted text is the original form.

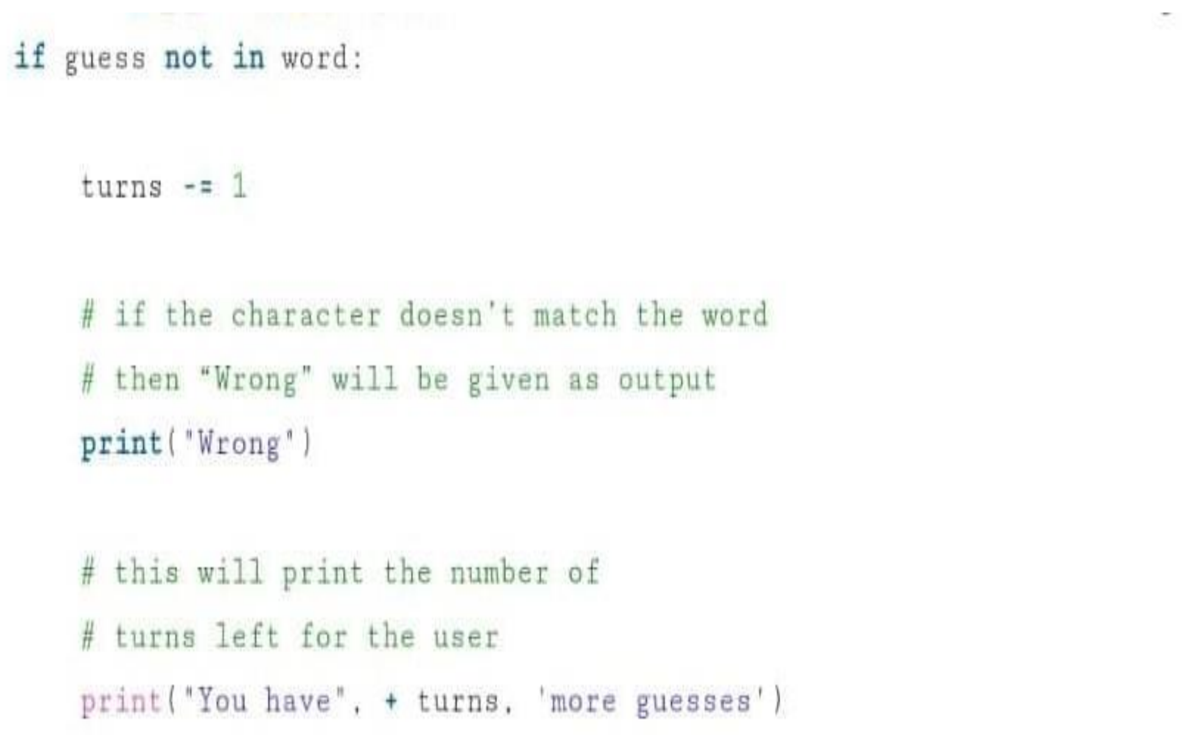
ANSWERS

Answer no 1:

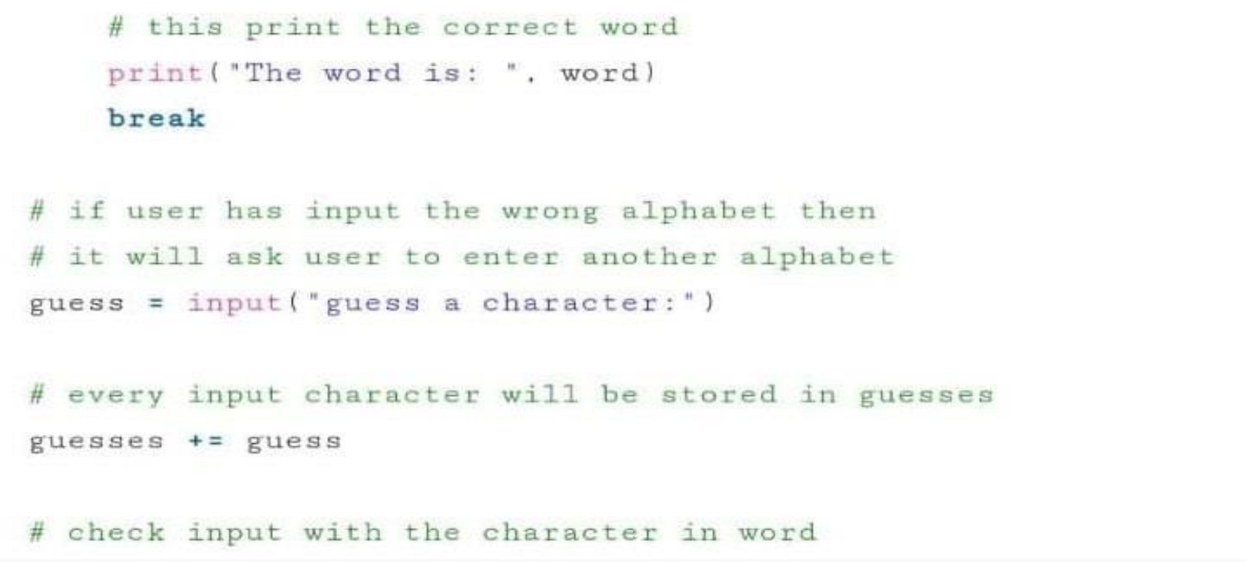


Below is the Python implementation:

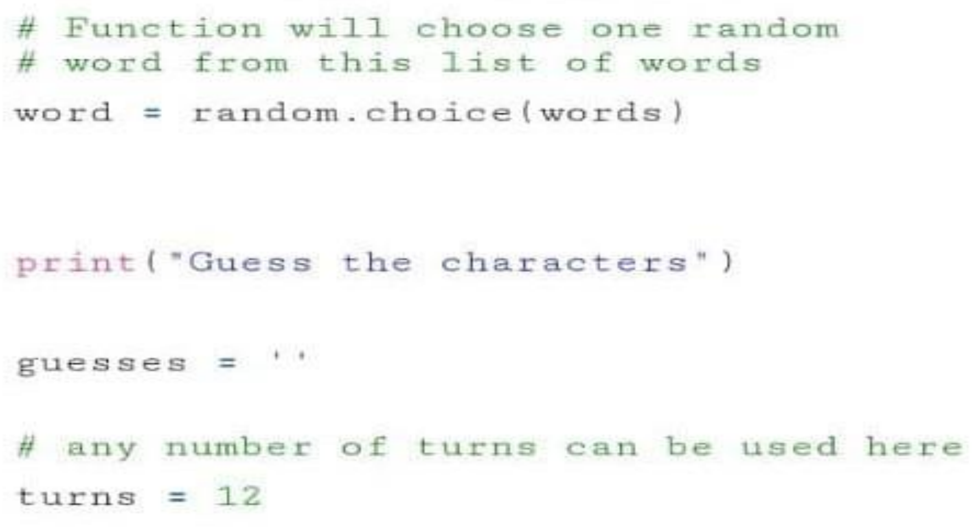
```
import random
# library that we use in order to choose
# on random words from a list of words

name = input("What is your name? ")
# Here the user is asked to enter the name first

print("Good Luck ! ", name)

words = ['rainbow', 'computer', 'science', 'programming',
         'python', 'mathematics', 'player', 'condition',
```



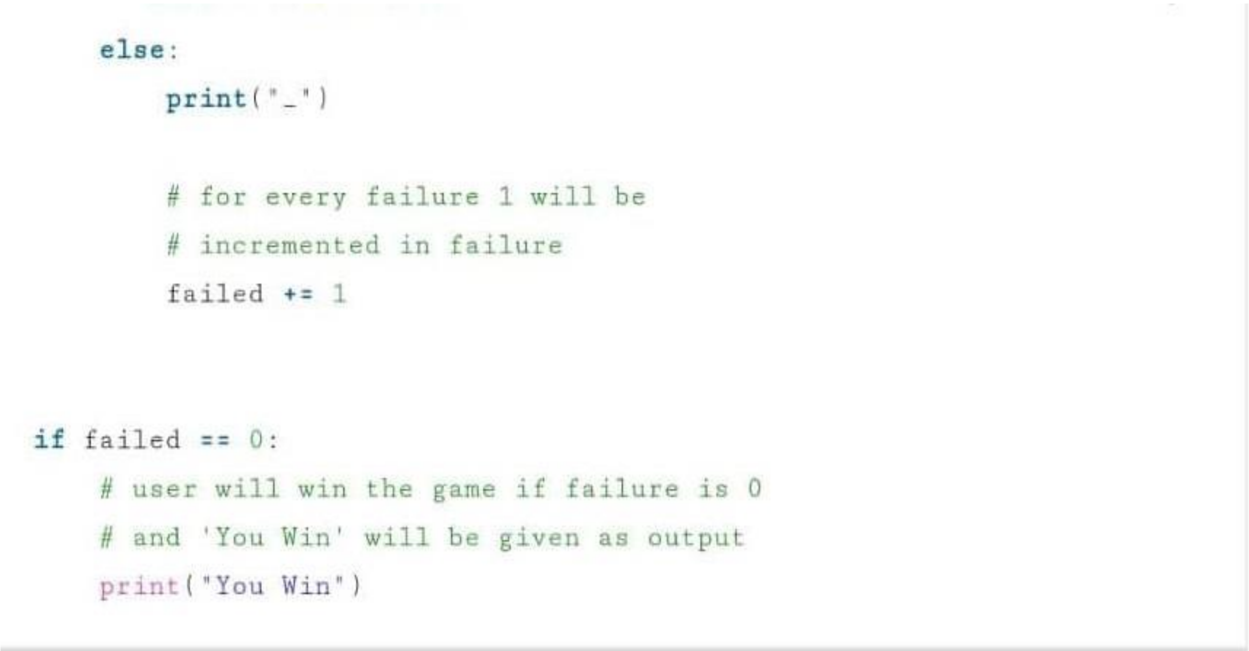
```
if guess not in word:

    turns -= 1

    # if the character doesn't match the word
    # then "Wrong" will be given as output
    print("Wrong")

    # this will print the number of
    # turns left for the user
    print("You have", + turns, 'more guesses')
```



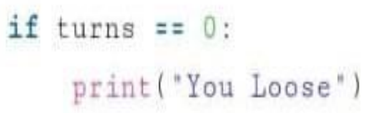
```
    # this print the correct word
    print("The word is: ", word)
    break

# if user has input the wrong alphabet then
# it will ask user to enter another alphabet
guess = input("guess a character:")

# every input character will be stored in guesses
guesses += guess

# check input with the character in word
```



```
# Function will choose one random
# word from this list of words
word = random.choice(words)


print("Guess the characters")

guesses = ''

# any number of turns can be used here
turns = 12
```

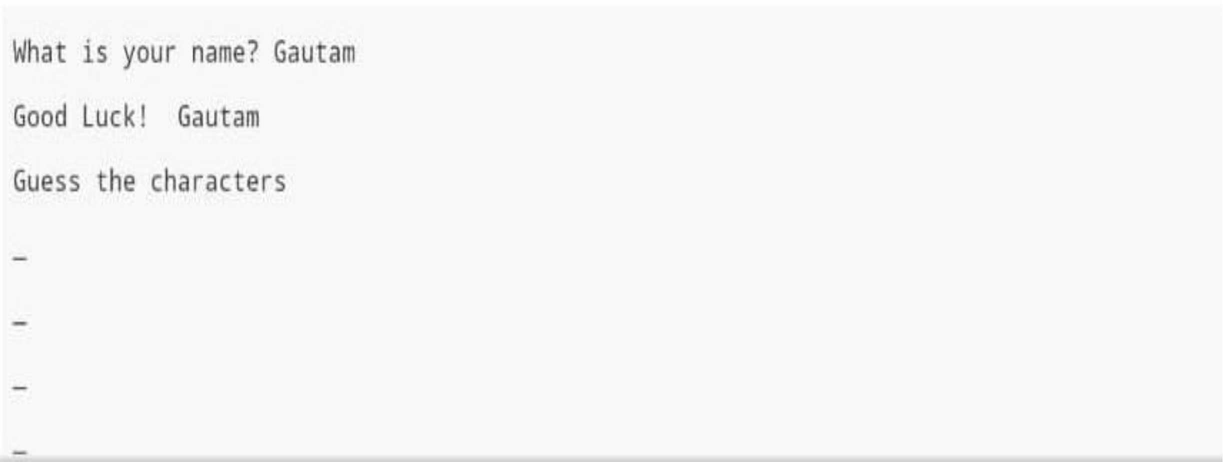


```
    else:
        print("_")

        # for every failure 1 will be
        # incremented in failure
        failed += 1


if failed == 0:
    # user will win the game if failure is 0
    # and 'You Win' will be given as output
    print("You Win")
```



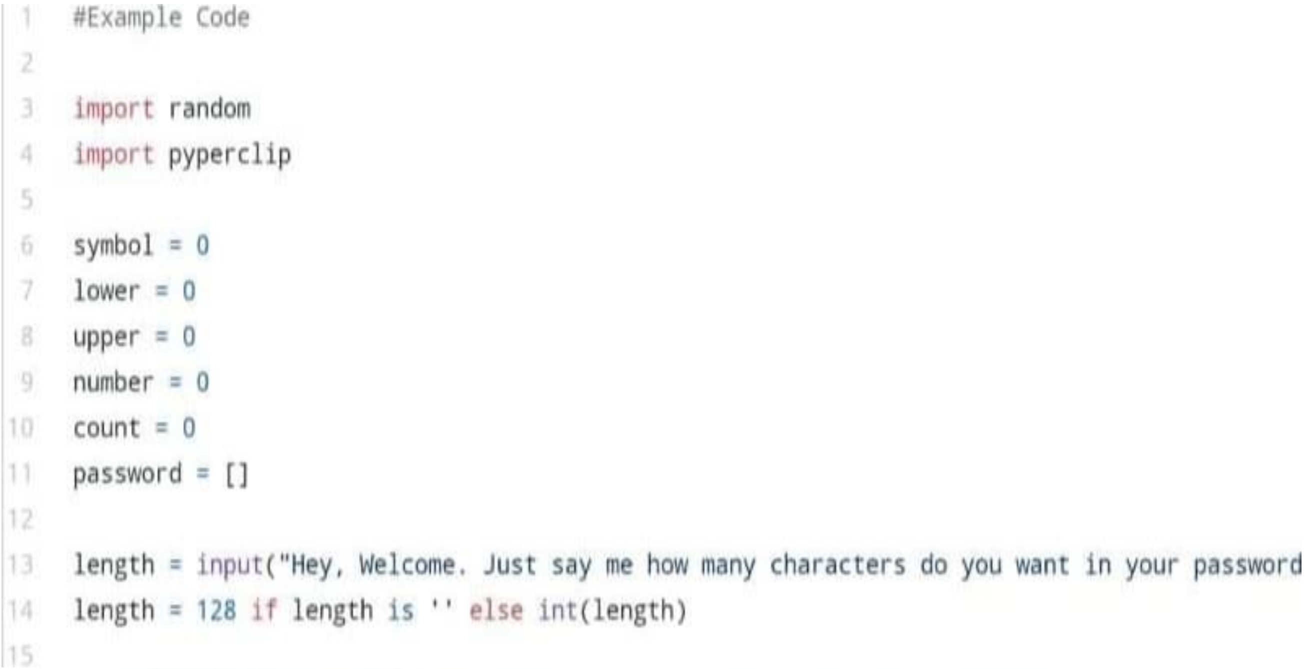
```
if turns == 0:
    print("You Loose")
```

Output:

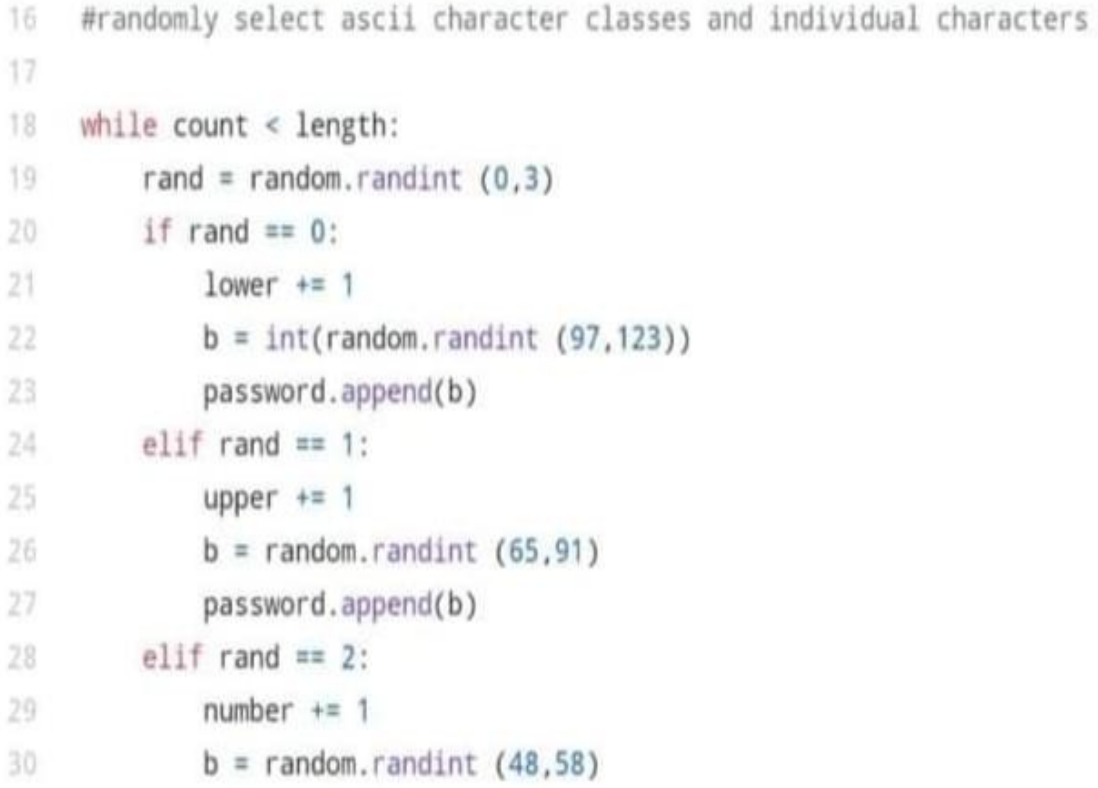


```
What is your name? Gautam

Good Luck!  Gautam

Guess the characters

_

_

_

_
```

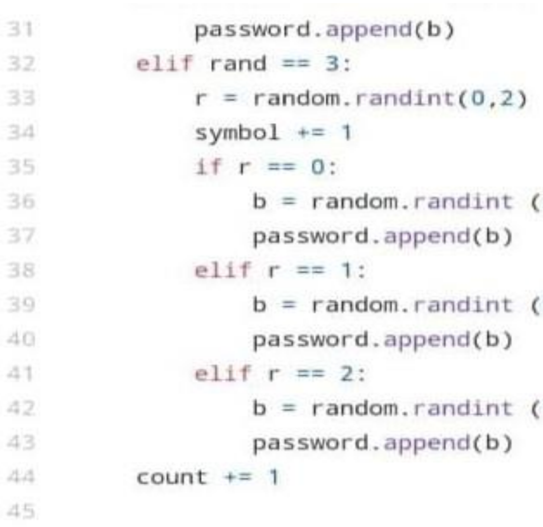
<u>Answer no 2</u>:



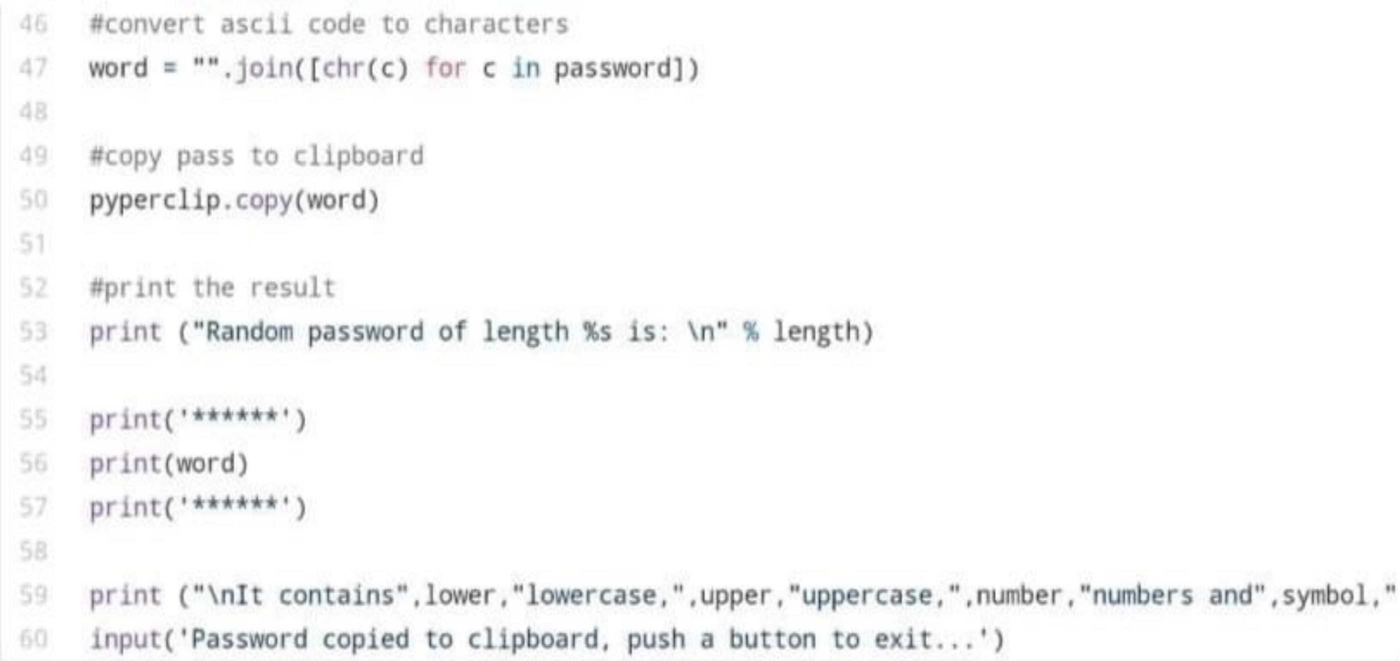
```
#Example Code

import random
import pyperclip

symbol = 0
lower = 0
upper = 0
number = 0
count = 0
password = []

length = input("Hey, Welcome. Just say me how many characters do you want in your password
length = 128 if length is '' else int(length)

```



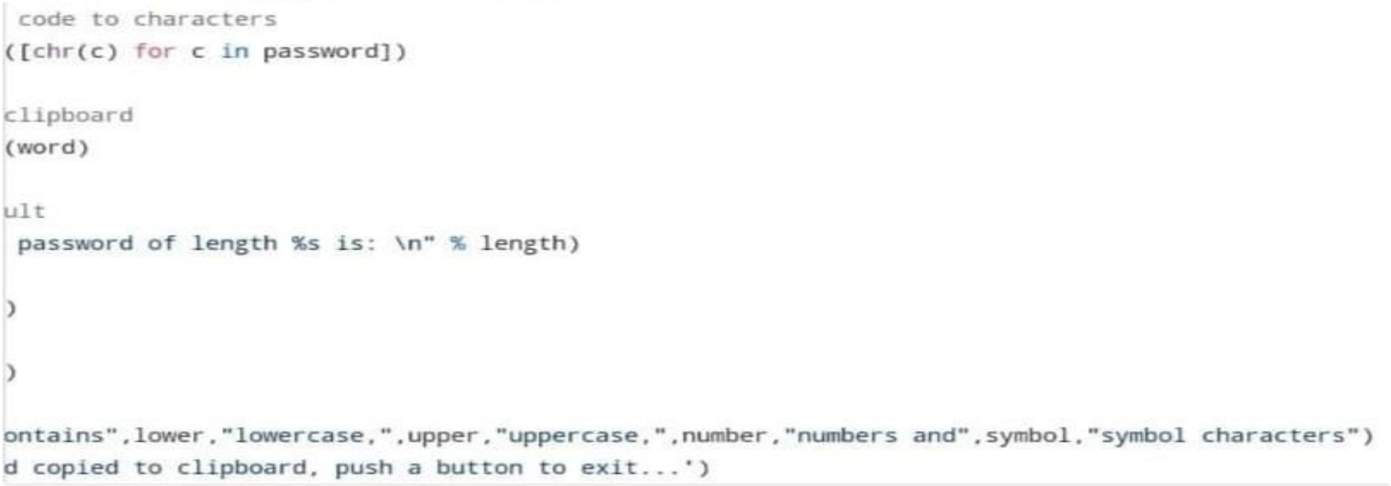
```
#randomly select ascii character classes and individual characters

while count < length:
    rand = random.randint (0,3)
    if rand == 0:
        lower += 1
        b = int(random.randint (97,123))
        password.append(b)
    elif rand == 1:
        upper += 1
        b = random.randint (65,91)
        password.append(b)
    elif rand == 2:
        number += 1
        b = random.randint (48,58)
```



```
            password.append(b)
    elif rand == 3:
        r = random.randint(0,2)
        symbol += 1
        if r == 0:
            b = random.randint (33,48)
            password.append(b)
        elif r == 1:
            b = random.randint (91,97)
            password.append(b)
        elif r == 2:
            b = random.randint (123,126)
            password.append(b)
    count += 1

```

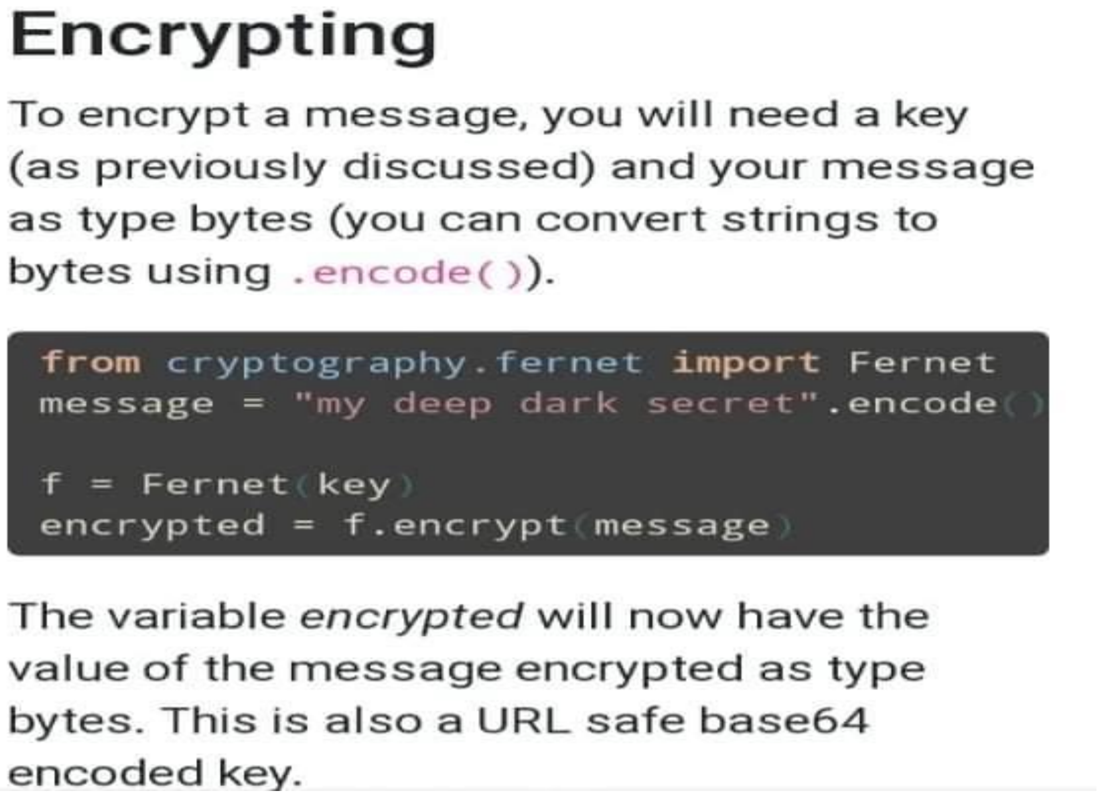


```
#convert ascii code to characters
word = "".join([chr(c) for c in password])

#copy pass to clipboard
pyperclip.copy(word)

#print the result
print ("Random password of length %s is: \n" % length)

print('******')
print(word)
print('******')

print ("\nIt contains",lower,"lowercase,",upper,"uppercase,",number,"numbers and",symbol,"
input('Password copied to clipboard, push a button to exit...')
```



```
 code to characters
([chr(c) for c in password])

clipboard
(word)

ult
 password of length %s is: \n" % length)

)

)

ontains",lower,"lowercase,",upper,"uppercase,",number,"numbers and",symbol,"symbol characters")
d copied to clipboard, push a button to exit...')
```

<u>Answer no 3</u>:

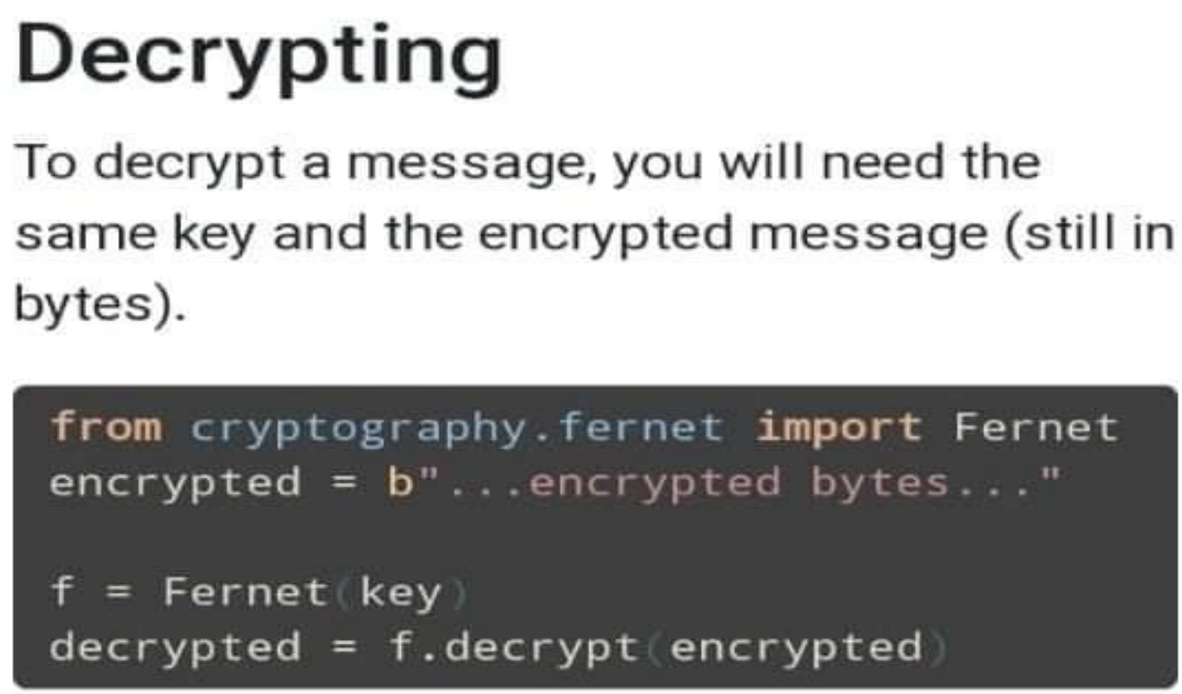


Encrypting

To encrypt a message, you will need a key (as previously discussed) and your message as type bytes (you can convert strings to bytes using `.encode()`).

```
from cryptography.fernet import Fernet
message = "my deep dark secret".encode()

f = Fernet(key)
encrypted = f.encrypt(message)
```

The variable *encrypted* will now have the value of the message encrypted as type bytes. This is also a URL safe base64 encoded key.



Decrypting

To decrypt a message, you will need the same key and the encrypted message (still in bytes).

```
from cryptography.fernet import Fernet
encrypted = b"...encrypted bytes..."

f = Fernet(key)
decrypted = f.decrypt(encrypted)
```

```
from cryptography.fernet import Fernet
key = b'' # Use one of the methods to ge
input_file = 'test.encrypted'
output_file = 'test.txt'

with open(input_file, 'rb') as f:
    data = f.read()

fernet = Fernet(key)
encrypted = fernet.decrypt(data)

with open(output_file, 'wb') as f:
    f.write(encrypted)

# You can delete input_file if you want
```

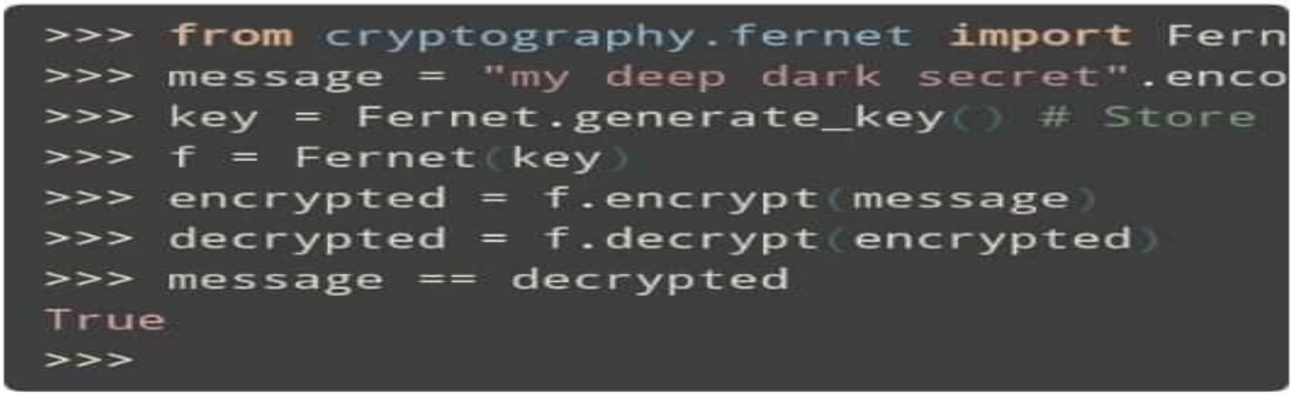
Encrypting and Decrypting Files

We can also encrypt files using this method since files can be read as bytes. Simply open the file, read the bytes, encrypt the data and write them out to a new file. To encrypt:

```
from cryptography.fernet import Fernet
key = b'' # Use one of the methods to ge
input_file = 'test.txt'
output_file = 'test.encrypted'

with open(input_file, 'rb') as f:
    data = f.read()

fernet = Fernet(key)
encrypted = fernet.encrypt(data)

with open(output_file, 'wb') as f:
    f.write(encrypted)

# You can delete input_file if you want
```

Demonstration

To show this in action, here is a properly constructed example.



```
>>> from cryptography.fernet import Fern
>>> message = "my deep dark secret".enco
>>> key = Fernet.generate_key() # Store
>>> f = Fernet(key)
>>> encrypted = f.encrypt(message)
>>> decrypted = f.decrypt(encrypted)
>>> message == decrypted
True
>>>
```

This example shows a key being generated, you will want to make sure you have already sorted your key out and put it in a file for later use.