

**Department of Electrical Engineering**  
**Assignment**

**Date: 07/05/2020**

---

**Course Details**

<b>Course Title:</b>	<u>Programing Fundamentals</u>	<b>Module:</b>	<u>08</u>
<b>Instructor:</b>	<u>Engr. Muhammad Waqas sir</u>	<b>Total</b>	<u>20</u>
		<b>Marks:</b>	

---

**Student**  
**Details**

<b>Name:</b>	<u>FAWAD AHMAD</u>	<b>Student ID:</b>	<u>13204</u>
--------------	--------------------	--------------------	--------------

---

Q1.	(a)	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.	Marks 5
			CLO 2
Q2.	(a)	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	Marks 10
			CLO 1
Q3.	(a)	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.	Marks 10
			CLO 1

### Question 1 (A)

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.

### Answer:

```
import random
words = ['fawadahmad', 'latif', 'inu',]
word = random.choice(words)
print("Guess the characters")
guesses = ""
turns = 4
while turns > 0:
    failed = 0
    for char in word:
        if char in guesses:
            print(char)
        else:
            print("_")
            failed += 1
    if failed == 0:
        print("You Win")
        print("The word is: ", word)
        break
    guess = input("guess a character:")
    guesses += guess
    if guess not in word:
        turns -= 1
        print("Wrong")
        print("You have", + turns, 'more guesses')
    if turns == 0:
        print("You Loose")
```

## Q1. INPUT

```
main.py  saved
1  import random
2  words = ['fawadahmad', 'latif', 'inu',]
3  word = random.choice(words)
4  print("Guess the characters")
5  guesses = ''
6  turns = 4
7  while turns > 0:
8      failed = 0
9      for char in word:
10         if char in guesses:
11             print(char)
12         else:
13             print("_")
14             failed += 1
15     if failed == 0:
16         print("You Win")
17         print("The word is: ", word)
18         break
19     guess = input("guess a character:")
20     guesses += guess
21     if guess not in word:
22         turns -= 1
23         print("Wrong")
24         print("You have", + turns, 'more guesses')
25         if turns == 0:
26             print("You Loose")
27
```

## OUTPUTS

```
Guess the characters
```

```
-
```

```
-
```

```
-
```

```
-
```

```
-
```

```
-
```

```
-
```

```
-
```

```
-
```

```
-
```

```
guess a character:fawadahmad
```

```
f
```

```
a
```

```
w
```

```
a
```

```
d
```

```
a
```

```
h
```

```
m
```

```
a
```

```
d
```

```
You Win
```

```
The word is: fawadahmad
```

```
➤ 
```

Guess the characters

-

-

-

guess a character:latif

Wrong

You have 3 more guesses

i

-

-

guess a character:fawadahmad

Wrong

You have 2 more guesses

i

-

-

guess a character:inu

i

n

u

You Win

The word is: inu

➤

## Question 2 (A)

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

## Answer:

```
import random
characters = ["A", "B", "C", "D", "E", "F", "G", "H", "I", 'J', 'K', 'L', 'M', 'N', 'O', 'P', 'Q', 'R', 'S', 'T',
'U', 'V', 'W', 'X', 'Y', 'Z', 'a', 'b', 'c', 'd', 'e', 'f', 'g', 'h', 'i', 'j', 'k', 'l', 'm', 'n', 'o', 'p', 'q', 'r', 's', 't', 'u', 'v', 'w',
'x', 'y', 'z', '1', '2', '3', '4', '5', '6', '7', '8', '9', '0', '!', '@', '#', '$', '%', '&', '*', '(', ')']
strength = input("how many letter password do you want?: ").lower()
new_password = []
def password(strength):
    if strength == 'eight':
        while len(new_password) != 8:
            new_password.append(characters[random.randint(1, 70)])
    elif strength == 'nine':
        while len(new_password) != 9:
            new_password.append(characters[random.randint(1, 70)])
    elif strength == 'ten':
        while len(new_password) != 10:
            new_password.append(characters[random.randint(1, 70)])
    elif strength == 'eleven':
        while len(new_password) != 11:
            new_password.append(characters[random.randint(1, 70)])
    elif strength == 'twelve':
        while len(new_password) != 12:
            new_password.append(characters[random.randint(1, 70)])
    elif strength == 'thirteen':
        while len(new_password) != 13:
            new_password.append(characters[random.randint(1, 70)])
    elif strength == 'fourteen':
        while len(new_password) != 14:
            new_password.append(characters[random.randint(1, 70)])
    return new_password
password(strength)
New_password = (new_password)
print(New_password)
```

## Q2. INPUT

```
main.py saved
1 import random
2
3 characters = ["A", "B", "C", "D", "E", "F", "G", "H", "I", "J", "K", "L", "M", "N", "O", "P",
              'Q', 'R', 'S', 'T', 'U', 'V', 'W', 'X', 'Y', 'Z', 'a', 'b', 'c', 'd', 'e', 'f', 'g', 'h', 'i',
              'j', 'k', 'l', 'm', 'n', 'o', 'p', 'q', 'r', 's', 't', 'u', 'v', 'w', 'x', 'y', 'z', '1', '2',
              '3', '4', '5', '6', '7', '8', '9', '0', '!', '@', '#', '$', '%', '&', '*', '(', ')']
4 strength = input("how many letter password do you want?: ").lower()
5 new_password = []
6 def password(strength):
7     if strength == 'eight':
8         while len(new_password) != 8:
9             new_password.append(characters[random.randint(1, 70)])
10    elif strength == 'nine':
11        while len(new_password) != 9:
12            new_password.append(characters[random.randint(1, 70)])
13    elif strength == 'ten':
14        while len(new_password) != 10:
15            new_password.append(characters[random.randint(1, 70)])
16    elif strength == 'eleven':
17        while len(new_password) != 11:
18            new_password.append(characters[random.randint(1, 70)])
19    elif strength == 'twelve':
20        while len(new_password) != 12:
21            new_password.append(characters[random.randint(1, 70)])
22    elif strength == 'thirteen':
23        while len(new_password) != 13:
24            new_password.append(characters[random.randint(1, 70)])
25    elif strength == 'fourteen':
```

```
24         new_password.append(characters[random.randint(1, 70)])
25     elif strength == 'fourteen':
26         while len(new_password) != 14:
27             new_password.append(characters[random.randint(1, 70)])
28     elif strength == 'fifteen':
29         while len(new_password) != 15:
30             new_password.append(characters[random.randint(1, 70)])
31     return new_password
32 password(strength)
33 New_password = (new_password)
34 print(New_password)
35
```

## OUTPUTS

01

```
how many letter password do you want?: eight
['2', 'w', '5', '*', 'J', 't', 'r', 'e']
> []
```

02

```
how many letter password do you want?: twelve
['D', '4', 'Q', 'P', 'O', 's', 'l', 'U', '2', '&', 'Q', 'u']
> []
```

03

```
how many letter password do you want?: fifteen
['s', 'g', 'H', '$', '2', 'J', 'o', 'b', 'W', 'D', 'u', 'L', '$', 's', 'j']
> []
```



### Question 3 (A)

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.

### Answer:

#### For encryption:

#### Program

```
def encrypt(message):
    newS=""
    for car in message:
        newS=newS+chr(ord(car)+2)
    return newS

print(encrypt('FAWAD AHMAD'))
```

### INPUT

```
main.py  saved
1  def encrypt(message):
2      newS=""
3      for car in message:
4          newS=newS+chr(ord(car)+2)
5      return newS
6
7
8  print(encrypt('FAWAD AHMAD'))
9
10
```

## Output of encryption

```
HCYCF"CJOCF
[]
```

## For Decryption:

```
def decrypt(message):
    newS=""
    for car in message:
        newS=newS+chr(ord(car)-2)
    return newS
```

```
print(decrypt(' HCYCF"CJOCF'))
```

## Decryption input

```
main.py saved
1 def decrypt(message):
2     newS=""
3     for car in message:
4         newS=newS+chr(ord(car)-2)
5     return newS
6
7
8 print(decrypt('HCYCF"CJOCF'))
9
10
11
12
13
```

## Output of decryption



---

END OF PAPER