Assignment 2

Date : / /20

Title :- Write python code for Guessing game / Hangman.

Problem Statement :- To python code for Guessing game / Hangman.

Pre-lab :- A basic understanding of computer programming technologies. A basic understanding of any of the programming languages will help in understanding python programming & datascience concepts.

Theory :-

This script / program is an interactive is an interactive guessing game, which will ask the user to guess a number between 1 & 99. We are using random module with radint function to get a random number. The script also contains a while loop which make script run until user guess right number

This is a python script of the classic game "Hangman". The word to guess is represented by a row of dashes. If Player guess a letter which exists in the word. the script writes it in all its correct positions. The player has 10 turns to guess the word. You can easily customize game by changing variables.

Pooja

Date : / / 20

Decisions -
The if-else is used to make choices in python code. This is a compound statement the simplest form is

```
if condition:
    action-1
else:
    action-2
```

The indentation is required. Not that else & its action are optional. The condition is an expression which evaluates to True or false.

```
X = 1
if X > 0:
  Print "Friday is wonderfull"
else:
  Print "Monday sucks"
Print "Have a good weekend"
```

Results in output
Friday is wonderfull
Have a good weekend.

When we change X = 0 then output will be
Monday sucks.
Have a good weekend.

Pooja

Date : / / 20

More complex decisions have several alternatives depending on several conditions. For these the elif is used. It means "else if" & one can have any number of elif clauses bet" the if & else.

```
if x>=0 and x<10:
    digits 1
if x >= 10 and x<100:
    digits 2.
if x>=100 and x<1000:
    digits 3
if x>=1000 and x<10000:
    digits =4.
else:
    digits =0      # more than four.
```

Loops -
else in loops.
A loop may have an optional else which is executed when loop finishes. for ex - the loop for n in [10,9,8,7,6,5,4,3,2,1]:

```
        print n,
    else:
        print "blast off"
```

O/P → 10 9 8 7 6 5 4 3 2 1 blastoff.

& the loop.

Date : / /20

```
n=10
while n>0 :
    print n,
    n=n-1
else :
    Print "blastoff"
```

has the same effect

Break, continue & pass

The break stmt like in C, breaks out smallest enclosing for or while loop. The continue stmt, continues with next iteration of loop. The pass stmt does nothing. It can be used when a stmt required syntactically but program requires no action.

Here is an ex-

```
for n in range (2,10):
    for x in range (2,n):
        if n%x==0:
            print n,'equals',x,'*',n/x
            break.
    else:
        # loop fell through without finding
            factor print n,' is a prime
                number.
```

O/P- 2 is a prime number.
3 is a prime number.
4 Equals 2 * 2

Date : / / 20

5 is a prime number
6 equals 2 * 3
7 is a prime number.
8 equals 2 * 4
9 equals 3 * 3.

Lists -
In python, lists are not required to be homogeneous.
a = [2, "jack", 45, "23 Wentworthe Ave"].
>>> a
[2, 'jack', 45, '23 wentworth Ave']
>>> a[0]
2
>>a[1]
'jack'.
>>> a[2]
45

Length of a list; empty list.
Every list has a length, number of items in list, obtained using len function.
>>> x = [9, 4, 900, -45]
>>> leng(x)
4.
>>> x = []
>>> leng(x)
0

Date : / /20

Sublists (slicing) -
start ≤ i < end.

```
x = range(0, 20, 2)
>>> x
[0, 2, 4, 6, 8, 10, 12, 14, 16, 18]
>>> x[2:5]
[4, 6, 8]
>>> x[0:5]
[0, 2, 4, 6, 8]
>>> x[:5]
[0, 2, 4, 6, 8].
>>> x[2:]
[4, 6, 8, 10, 12, 14, 16, 18]
```

joining two lists -

```
>>> [2, 3, 6, 10] + [4, 0, 0, 5, 0].
[2, 3, 6, 10, 4, 0, 0, 5, 0].
```

List methods -
x.append (item)

```
>>> x = [3, 6, 8, 9]
>>> x.append (999)
>>> x
[3, 6, 8, 9, 999].

>>> x = ['a', 'c', '3', 'd', '7']
>>> x.insert (0, 100)
>>> x
[100, 'a', 'c', '3', 'd', '7'].
```


Pooja

Date : / / 20

```
>>> x.remove('a')
>>> x
[100, 'c', 'junk', '3', 'd', 7]

>>> x.pop(0)
100
>>> x
['c', 'junk', '3', 'd', 7]

>>> x.pop()
7
>>> x
['c', 'junk', '3', 'd']
```

Strings :-

```
>>> x = 'gobbletygook'
>>> x[2]
'b'
>>> x[5]
'e'
>>> x[5] = 's'
```

Traceback.

strings are immutable.

```
>>> a = 'gobb is refreshing'
>>> a.capitalize()
'Gobb is refreshing'
```

Date : / /20 /20

Algorithm : Guessing game.

```
import random
n = random.randint(1,99)
guess = int(raw_input("Enter integer 1 to 99"))
while n != "guess":
    print
    if guess < n:
        print "guess is low"
        guess = int(raw_input("Enter integer
                    from 1 to 99"))
    elif guess > n:
        print "guess is high"
        guess = int(raw_input("Enter integer
                    from 1 to 99"))
    else:
        print "you guessed it!"
        break.
    print
```

Algorithm 2 Hangman.

```
# importing the time module
import time.
# welcoming user.
name = raw_input("what is your name")
print "Hello," + name, "Time to play"
print ""
# wait for 1 second.
```

Date : / / 20

```
time.sleep(1)
Print "start guessing...."
time.sleep(0.5)

#here we set the secret
 word = "secret"

#creates an variable with an empty value
guesses = ''
#determine number of turns.
turns = 10

#creat a while loop:
#check if turns are more than 0.
 while turns > 0:
    #mak a counter that starts with zero.
    failed = 0

    #for every character in secret_word.
    for char in word:

    #see if character is in players guess
      if char in guesses:

      #print then out character.
        print char,

      else
        Print "_",
```


Pooja

```
# and increase failed counter with 1.
    failed += 1.

# if failed is equal to zero.
# print you won.
  if failed == 0;
     print "you won"

# exit the script
  break.

Print.

guess = raw_input (" guess a character")
guesset = guess.
if guess not found in word.
    turns -= 1.
    print "wrong"

    print "you have", +turns, 'more guesses'

    if turns == 0:

        print "you lose"
```

Post lab - students will be able to execute various python commands on python shell as well as using editor for their application.

Date : / / 20

Conclusion :- Thus student can implement this game by using Condition Statement.