python_basic_programming_16

May 20, 2023

```
[ ]: 1.Write a function that stutters a word as if someone is struggling to read it.
     The first two letters are repeated twice with an ellipsis ... and space after
      ↪each,
     and then the word is pronounced with a question mark ?
     Examples: stutter("incredible") ⫾ "in... in... incredible?"
     stutter("enthusiastic") ⫾ "en... en... enthusiastic?"
     stutter("outstanding") ⫾ "ou... ou... outstanding?"

     Hint :- Assume all input is in lower case and at least two characters long.
```

```
[1]: def stutterWord():
         in_string = input('Enter the Word :')
         out_string = in_string.replace(in_string[0:2],((in_string[0:2]+'... ')*2)+
      ↪in_string[0:2])  +'?'
         print(f'{in_string}   {out_string}')

     for i in range(3):
         stutterWord()
```

```
Enter the Word :incredible
incredible   in… in… incredible?
Enter the Word :enthusiastic
enthusiastic   en… en… enthusiastic?
Enter the Word :outstanding
outstanding   ou… ou… outstanding?
```

```
[ ]: 2..Create a function that takes an angle in radians and returns the
      ↪corresponding angle in degrees
     rounded to one decimal place ?
     Examples: radians_to_degrees(1) ⫾ 57.3
     radians_to_degrees(20) ⫾ 1145.9
     radians_to_degrees(50) ⫾ 2864.8
```

```
[2]: import math
     def radianToDegree():
         in_num = int(input('Enter the angle in Radians: '))
         out_num = (180/math.pi)*in_num
         print(f'{in_num} radian(s)   {out_num:.1f} degrees')
```

```
for x in range(3):
    radianToDegree()
```

```
Enter the angle in Radians: 1
1 radian(s)   57.3 degrees
Enter the angle in Radians: 20
20 radian(s)   1145.9 degrees
Enter the angle in Radians: 50
50 radian(s)   2864.8 degrees
```

[ ]: 3.In this challenge, establish if a given integer num is a Curzon number.
If 1 plus 2 elevated to num is exactly divisible by 1 plus 2 multiplied by num,
then num is a Curzon number. Given a non-negative integer num,
implement a function that returns True if num is a Curzon number, or False␣
↪otherwise.
Examples: is_curzon(5) ▌ True # 2 ** 5 + 1 = 33 # 2 * 5 + 1 = 11 # 33 is a␣
↪multiple of 11 is_curzon(10)   False # 2 ** 10 + 1 = 1025 # 2 * 10 + 1 = 21 #␣
↪1025 is not a multiple of 21 is_curzon(14)   True # 2 ** 14 + 1 = 16385 # 2 *␣
↪14 + 1 = 29 # 16385 is a multiple of 29

[3]:
```python
def checkCurzon():
    in_num = int(input("Enter a number: "))
    if (pow(2,in_num)+1)%((2*in_num)+1) == 0:
        print(f'{in_num} is a Curzon Number')
    else:
        print(f'{in_num} is Not a Curzon Number')

for x in range(4):
    checkCurzon()
```

```
Enter a number: 5
5 is a Curzon Number
Enter a number: 10
10 is Not a Curzon Number
Enter a number: 12
12 is Not a Curzon Number
Enter a number: 14
14 is a Curzon Number
```

[ ]: 4.Given the side length x find the area of a hexagon ?
Examples: area_of_hexagon(1) ▌ 2.6
area_of_hexagon(2) ▌ 10.4
area_of_hexagon(3) ▌ 23.4

[5]:
```python
import math
def areaOfHexagon():
    in_num = int(input('Enter the side length of a Hexagon: '))
```

```
        out_num = ((3*math.sqrt(3))/2)*(pow(in_num,2))
        print(f'Area for Hexagon of sidelength {in_num}   {out_num:.1f}')

for x in range(3):
    areaOfHexagon()
```

```
Enter the side length of a Hexagon: 2
Area for Hexagon of sidelength 2   10.4
Enter the side length of a Hexagon: 10
Area for Hexagon of sidelength 10   259.8
Enter the side length of a Hexagon: 14
Area for Hexagon of sidelength 14   509.2
```

[ ]: 5.Create a function that returns a base-2 (binary) representation of a base-10
↳(decimal) string number. To convert is simple:
((2) means base-2 and (10) means base-10)
010101001(2) = 1 + 8 + 32 + 128.
Going from right to left, the value of the most right bit is 1, now from that
↳every bit
to the left will be x2 the value, value of an 8 bit binary numbers are (256,
↳128, 64, 32, 16, 8, 4, 2, 1).
Examples:
binary(1) ⎮ "1" # 1* 1 = 1 binary(5)   "101" # 1 1 + 1 4 = 5 binary(10)   "1010"
↳# 1 2 + 1 8 = 10

[6]:
```
def getBinary():
    in_num = int(input("Enter a Number: "))
    out_num = bin(in_num).replace('0b','')
    print(f'Binary of {in_num}   {out_num}')

for x in range(3):
    getBinary()
```

```
Enter a Number: 1
Binary of 1   1
Enter a Number: 4
Binary of 4   100
Enter a Number: 12
Binary of 12   1100
```