Parson Puzzles [Solutions]

1. Read in a number and write out whether it is odd or even

```
num = input("Enter a number: ")
mod = num % 2
if mod > 0:
    print("You picked an odd number.")
else:
    print("You picked an even number.")
```

2.Guessing game 1 to 9

```
import random
rd = random.randint(1,9)
guess = int( input("Enter a guess between 1 to 9") )
c = 1
while guess != rd and guess != "exit":
    if guess == rd:
        print("Right!")
        print("You took only", c, "tries!")
    else:
        if guess > rd:
            print("Too high"
        else:
            print("Too low")
        guess = int( input("Enter a guess between 1 to 9") )
        c += 1
```

3. Read a file and print out the length of line with a space between each length.

```
f = open('filename.txt', 'r')
lines = f.readlines()
for line in lines:
    length = len( line )
    print( length, end = " "  )
f.close()
```

4. Given a list of words, both write the words backwards and order the words in the reverse direction

```
words = ["once","upon","a","time","in","oppositeland"]
neworder = []
for word in words:
    new_word = ""
    for letter in word:
        new_word = letter + new_word
```

```
      1 neworder = [new_word] + neworder
4 print(neworder)
```

5. Given a dictionary named `heights` with string names as keys and heights (in meters) as values, print tallest.

```
3 max = 0
7 tallest = 'No-one'
1 for person in heights:
      6 height = heights[ person ]
      5 if  height > max:
            8 max = height
            4 tallest  = person
2 print( 'the tallest person is ', tallest )
```

6. Given a file named `data.csv` with names as strings and scores as numbers in the range 0 to 100 inclusive, print lowest

```
2  file = open( 'data.csv' )
4  name = 'No-one'
5  min = 100
10 for line in file:
      9  score = int( ( line.split( ' ,' ) )[ 1 ] )
      1  if score < min:
            11 min = score
            3  name = ( line.split( ',' ) )[ 0 ]
            8  minName = name
6  print( 'person with min score is' + minName )
7  file.close()
```

7. Write a program to read the data into a list of dictionaries, each dictionary holding a name and an age for a single person.

```
5 f = open( "data.txt" )
3 lines = f.readlines()
8 people = []
7 for lineNum in range( 0, len( lines ), 3 ):
      9 newName = lines[ lineNum ][ :-1 ]
      6 newAgeAsString = lines[ lineNum + 1 ][ :-1 ]
      1 newAge  = int( newAgeAsString )
      2 newPerson = { "name" : newName, "age" : newAge }
      4 people = people + [ newPerson ]
```

8. Parson's Problem Generator: Write a program which takes another program in a text file, shuffles the line order, trims the leading whitespace and adds a line number to each reordered line.

```
11 from random import shuffle
1  f = open( "program.txt" )
10 lines = f.readlines()
```

```
8  shuffle(lines)
4  current_line = 1
2  for line in lines:
    9  while line[0] == " ":
        6  line = line[1:]
    5  line = str(current_line) + ". " + line[:-1]
    7  current_line += 1
    3  print(line)
```

9. Caesar Shift Cipher: Write a program that replaces each letter in a string with the next in the alphabet, ignoring whitespace

```
4  import string
8  alphabets=list(string.ascii_lowercase)
9  plaintext = "duel by dawn"
11 withoutblank = ""
12 for letter in plaintext:
     2  if letter != " ":
          1  withoutblank += letter
10 decoded = ""
7  for letter in withoutblank:
     5  i = alphabets.index(letter)
     3  decoded += alphabets[(i + 1)% len(alphabets)]
6  print(decoded)
```

10. (Challenge) Matrix multiplication: Given two 2D lists (forming matrices), compute the matrix multiplication

$$\begin{pmatrix} c_{11} & c_{12} \\ c_{21} & c_{22} \end{pmatrix} = \begin{pmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{pmatrix} \begin{pmatrix} b_{11} & b_{12} \\ b_{21} & b_{22} \end{pmatrix}$$

$$\begin{aligned} c_{11} &= a_{11}b_{11} + a_{12}b_{21} \\ c_{12} &= a_{11}b_{12} + a_{12}b_{22} \\ c_{21} &= a_{21}b_{11} + a_{22}b_{21} \\ c_{22} &= a_{21}b_{12} + a_{22}b_{22}. \end{aligned}$$

$$\begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix} \times \begin{bmatrix} 2 & 0 \\ 1 & 2 \end{bmatrix} = \begin{bmatrix} 4 & 4 \\ 10 & 8 \end{bmatrix}$$

$$\begin{bmatrix} 2 & 0 \\ 1 & 2 \end{bmatrix} \times \begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix} = \begin{bmatrix} 2 & 4 \\ 7 & 10 \end{bmatrix}$$

```
5  def matrixmultiplication (A, B):
     11 Arows = len(A)
     13 Acols = len(A[0])
     15 Brows = len(B)
     2  Bcols = len(B[0])
     3  if Acols != Brows:
          7  print("Not applicable")
          16 return
```

```
10 C = []
6  for row in range(0,Bcols):
       8  newrow = []
       1  for col in range(0,Arows):
              19 newrow += [0]
       17 C += [newrow]
9  for i in range(Arows):
       18 for j in range(Bcols):
              4  for k in range(Acols):
                     14 C[i][j] += A[i][k] * B[k][j]
12 return C

A = [[1,2],[3,4]]
B = [[2,0],[1,2]]
print(matrixmultiplication(A,B))
```