Python evaluation questions

1. Write a function which will take an integer as input and print each digit in a separate line. You are not allowed to use str or any other method will convert the integer into string.

```
Input: 1011

Output:
1
1
0
1
```

2. You are given words. Some words may repeat. For each word, output its number of occurrences. The output order should correspond with the input order of appearance of the word. See the sample input/output for clarification.
   Input format: two parameters: first number of words, second list of words to process
   Output format: Output two lines
   First line: Number of unique words
   Second line: Number of occurrences for each distinct word according to their appearance in the input.

```
Input:
4, [“bcdef”, “abcdefg”, “bcde”, “bcdef”]

Output:
3
2 1 1

Output explanation:

First output distinct words.
Second line shows the count for each word which occurred in the list.
```

3. You are given a positive integer. You can only use one **for loop and one print statement**. Print a numerical triangle of height like the one below:

```
Input:
5

Output:
1
22
333
4444
```

4. You are given a string. Suppose a character ‘c’ occurs 4 times in the string. Replace these consecutive occurrences of the character 'c' with (4, c) in the string.

Input format: A single line of input consisting of string s
Output format: A single line of output consisting of the modified string.

```
Input:
1222311

Output:
(1, 1) (3, 2) (1, 3) (2, 1)
```

5. You are given a N x M grid of strings. It consists of alphanumeric spaces and characters, spaces, and symbols (!,@,#,$,%,&). To decode a string, you need to read each column and select only alphanumeric characters and concatenate them. If there are symbols between two alphanumeric characters of the decoded script, then replace them with empty string ‘’. You need to do this without using the if condition.

```
Input:
7 3
Tsi
h%x
i #
sM
$a
#t%
ir!

Output
This is Matrix#  %!
```

6. You get an array of numbers, and return the sum of all of the positive ones. Example [1, -4, 7, 12] => 1+7+12 = 20. If there is nothing to sum return 0. You can not **use the if statement**.

7. Sum all the numbers of a given array (cq. list), except the highest and the lowest element (by value, not by index!). You can not **use the if statement.**

```
{ 6, 2, 1, 8, 10 } => 16

{ 1, 1, 11, 2, 3 } => 6
```

8. Our football team has finished the championship. Our team's match results are recorded in a collection of strings. Each match is represented by a string in the format `x:y`, where `x` is our team and `y` is our opponents score. The rules to calculate score is
   a. If x > y: 3 points
   b. If x < y: 0 point
   c. If x = y: 1 point

   We need to write a function that takes this collection and returns the number of points our team (`x`) got in the championship by the rules given above.

9. Complete the function that accepts a string parameter, and reverses each word in the string. All spaces in the string should be retained.

```
"This is an example!" ==> "sihT si na !elpmaxe"

"double  spaces"      ==> "elbuod  secaps"
```

10. You are going to be given a word. Your job is to return the middle character of the word. If the word's length is odd, return the middle character. If the word's length is even, return the middle 2 characters.

```
test => es
testing => t
A => A
```

11. You are given an array(list) strarr of strings and an integer k. Your task is to return the first longest string consisting of k consecutive strings taken in the array.

```
strarr = ["tree", "foling", "trashy", "blue", "abcdef", "uvwxyz"], k = 2

Concatenate the consecutive strings of strarr by 2, we get:

treefoling   (length 10)  concatenation of strarr[0] and strarr[1]
folingtrashy ("      12)  concatenation of strarr[1] and strarr[2]
trashyblue   ("      10)  concatenation of strarr[2] and strarr[3]
blueabcdef   ("      10)  concatenation of strarr[3] and strarr[4]
abcdefuvwxyz ("      12)  concatenation of strarr[4] and strarr[5]

Two strings are the longest: "folingtrashy" and "abcdefuvwxyz".
The first that came is "folingtrashy" so
longest_consec(strarr, 2) should return "folingtrashy".
```

12. Your task is to sort a given string. Each word in the string will contain a single number. This number is the position the word should have in the result. Note: Numbers can be from 1 to 9. So 1 will be the first word (not 0). If the input string is empty, return an empty string. The words in the input String will only contain valid consecutive numbers.

```
"is2 Thi1s T4est 3a"  -->  "Thi1s is2 3a T4est"
"4of Fo1r pe6ople g3ood th5e the2"  -->  "Fo1r the2 g3ood 4of th5e pe6ople"
""  -->  ""
```

13. Your task is to create a program which will play Rock Paper Scissors with the user. Take input from the user for his/her selection like scissors/rock/paper. Program should select randomly rock/paper/scissors. Output should be indicating who won the user or computer.

```
Input from user: scissors
Random selection from computer: paper
Output: User won

Input from user: rock
Random selection from computer: paper
Output: Computer won
```

14. Implement the function unique_in_order which takes as argument a sequence and returns a list of items without any elements with the same value next to each other and preserving the original order of elements.

```
unique_in_order('AAAABBBCCDAABBB') == ['A', 'B', 'C', 'D', 'A', 'B']
unique_in_order('ABBCcAD')         == ['A', 'B', 'C', 'c', 'A', 'D']
unique_in_order([1, 2, 2, 3, 3])   == [1, 2, 3]
```

15. ATM machines allow 4 or 6 digit PIN codes and PIN codes cannot contain anything but exactly 4 digits or exactly 6 digits. If the function is passed a valid PIN string, return true, else return false.

```
"1234"   -->  true
"12345"  -->  false
"a234"   -->  false
```

16. Complete the solution so that the function will break up camel casing, using a space between words.

```
"camelCasing"  =>  "camel Casing"
"identifier"   =>  "identifier"
""             =>  ""
```

17. Given a positive integer n written as abcd... (a, b, c, d... being digits) and a positive integer p. we want to find a positive integer k, if it exists, such that the sum of the digits of n taken to the successive powers of p is equal to k * n.

dig_pow(695, 2) should return 2 since $6^2 + 9^3 + 5^4 = 1390 = 695 * 2$
dig_pow(46288, 3) should return 51 since $4^3 + 6^4 + 2^5 + 8^6 + 8^7 = 2360688 = 46288 * 51$

18. Write a function that takes a string of parentheses, and determines if the order of the parentheses is valid. The function should return true if the string is valid, and false if it's invalid.

```
"()"              =>  true
")(()))"          =>  false
"("               =>  false
"(())((()())())"  =>  true
```

19. Write a function that accepts a fight string consisting of only small letters and return who wins the fight. When the left side wins, the Left side wins!, when the right side wins, the Right side wins!, in other cases, Let's fight again!.
Left side letters

```
w - 4
p - 3
b - 2
s - 1
t - 0 (pretty word)
```

Right side letters

```
m - 4
q - 3
d - 2
z - 1
j - 0 (pretty word)
```

The other letters don't have power and are only victims.
The priest units t and j change the adjacent letters from hostile letters to friendly letters with the same power.

```
alphabet_war("z")        #=>  "z"  => "Right side wins!"
```

Explanation:
Letter z belongs to the right side letters and the left side has no letter so the right side wins.

```
alphabet_war("tz")       #=>  "ts" => "Left side wins!"
```

Explanation:
Letter t is a pretty letter belonging to the left side. Pretty letters convert hostile letters to friendly letters with the same power. Z is a hostile letter with power 1. T will convert it to s a friendly letter with the same power.

20. Given two positive integers m (width) and n (height), fill a two-dimensional list (or array) of size m-by-n in the following way:
    a. All the elements in the first and last row and column are 1.
    b. All the elements in the second and second-last row and column are 2, excluding the elements from step 1.
    c. All the elements in the third and third-last row and column are 3, excluding the elements from the previous steps.
    d. And so on …

    Given m = 10, n = 9, your function should return

```
[[1, 1, 1, 1, 1, 1, 1, 1, 1, 1],
 [1, 2, 2, 2, 2, 2, 2, 2, 2, 1],
 [1, 2, 3, 3, 3, 3, 3, 3, 2, 1],
 [1, 2, 3, 4, 4, 4, 4, 3, 2, 1],
 [1, 2, 3, 4, 5, 5, 4, 3, 2, 1],
 [1, 2, 3, 4, 4, 4, 4, 3, 2, 1],
 [1, 2, 3, 3, 3, 3, 3, 3, 2, 1],
 [1, 2, 2, 2, 2, 2, 2, 2, 2, 1],
 [1, 1, 1, 1, 1, 1, 1, 1, 1, 1]]
```