Assignment 1:

Ques1:

In Scrabble1 each player has a set of tiles with letters on them. The object of the game is to use those letters to spell words. The scoring system is complex, but longer words are usually worth more than shorter words.
Imagine you are given your set of tiles as a string, like "quijibo", and you are given another string to test, like "jib".
Write a logic that takes two strings and checks whether the set of tiles can spell the word. You might have more than one tile with the same letter, but you can only use each tile once.

Ques2:

We have a string s, s = '12345' * 5.
Produce following ouput by using string slicing:
1. '11111'
2. '55555'
3. Reverse the string

Ques3:

Write a Python program to construct the following pattern, using a nested for loop.

```
*
* *
* * *
* * * *
* * * * *
* * * *
* * *
* *
*
```

Assignment 2:

Ques1:

Given a list of tuples, the task is to remove all tuples having duplicate first values from the given list of tuples.
Input: [('Tuple1', 121), ('Tuple2', 125), ('Tuple1', 135), ('Tuple4', 478)]
Output: [('Tuple1', 121), ('Tuple2', 125), ('Tuple4', 478)]

Ques2:

Given an array of names of candidates in an election. A candidate name in array represents a vote casted to the candidate. Print the name of candidates received Max vote. If there is a tie, print lexicographically smaller name.

Examples:

Input : votes[] = {"john", "johnny", "jackie", "johnny", "john", "jackie", "jamie", "jamie", "john", "johnny", "jamie", "johnny", "john"};
Output : John

We have four Candidates with name as 'John',
'Johnny', 'jamie', 'jackie'. The candidates
John and Johnny get maximum votes. Since John
is alphabetically smaller, we print it.

Assignment 3:

Ques1:

1) write a python program to create a text file containing countings of all the numbers from 1 to 20 in this format :
   2 : 2, 4, 6, 8, 10, 12, 14, 16, 18, 20
   3 : 3, 6, 9, 12, 15, 18, 21, 24, 27, 30… and so on.

2) write another script to append counting of next 20 numbers. I.re 21 - 40
3) Read the file created above and load it into dictionary in the format given below
   My_dictionary = {
   2 : [2, 4, 6, 8, 10….]
   3 : [3, 6, 9, 12...]
   }
   Note- Do not make dictionary using loop and multiplication. Make it using reading the text from the file and then use string methods.
4) Once done with creating the dictionary file then dump the dictionary object using pickle.
5) Write a python program to load the dictionary created above and then take user input and then print table list on console.

Assignment 4:

Ques1:

1. Create a python package with name *my_date* and create a module in this package with name *date_op.py*. Write a function for all of the following functionalities in this module:

my_date
-- date_operations.py
-- __init__.py
1. get_current_datetime
2. When entering age in string: 19 years, 5 months, 20 days. It should return birth_date
3. When entering birth_date. Return age in the above format.
4. When entering birth_date: Return how many time will person take to turn 50 years. Return string in above format.

2. Write a main.py file that import these functions and show the use of them.

Assignment 5:

Ques1:

Write an base class name Shape that have a abstract method: def area()
ALso have a class method name get() which accepts a name of the shape and returns the child class object

Child classes will implement how to compute the area. Take at least three child classes and show computed area for each shape

Ques2:

1. Create a super class called *Car*. The Car class has the following properties. Speed; regularPrice; color;
   It should have a method *doublegetSalePrice()* which will return the sale price of the car.
2. Create a sub class of Car class and name it as *Truck*. The Truck class has a field : weight.
   Override doublegetSalePrice() in a way that If weight>2000 then 10% discount, Otherwise no discount
3. Create a subclass of Car class and name it as *Ford*. The Ford class has a field manufacturer Discount.
   Override doublegetSalePrice() in a way that it will always subtract the manufacturer Discount.
4. Create a subclass of Car class and name it as *Sedan*. The Sedan class has a field length.
   Override doublegetSalePrice() in a way that If length > 20 feet, 5% discount, Otherwise , 10% discount.

Create instances of all type of cars and show the sale price of all of them.

Ques3:

Design a linked list that supports these ops:

1) Adds an element at the end.
2) Adds an element between the two elements.
3) Deletes an element from the given key.
4) Search an element from the given key.
5) Display the complete linked list.
6) Reverse the linked list.

Assignment 6:

Ques1:

Create a stack class with push and pop operations
Also create two custom exceptions one for Underflow condition and one for Overflow condition that will be raised when stack underflow or overflow condition occurs

Ques2:

Print all palindrome numbers from 10 to 999999 using generators.

Ques3:

We have a bookshop, in which orders are stored in format like this:
[

[order_number 1, (book number 1, quantity, price per unit), ... (book number 10, quantity, price per unit), [order_number 2, ……...]]

1. Write a program which returns a list of two tuples with (order number, total amount of order).
2. From this returns a list of order numbers for which total amount > x

Sample data:

```
[ [1, ("5464", 4, 9.99), ("8274",18,12.99), ("9744", 9, 44.95)],

  [2, ("5464", 9, 9.99), ("9744", 9, 44.95)],

  [3, ("5464", 9, 9.99), ("88112", 11, 24.99)],

  [4, ("8732", 7, 11.99), ("7733",11,18.99), ("88112", 5, 39.95)] ]
```

In 2nd point consider x to be `400`

Ques4:

We have a function named *calculate_sum(list)* which is calculating the sum of the list. Someone wants to modify the behaviour in a way such that:

1. It should calculate only sum of even numbers from the given list.
2. Obtained sum should always be odd. If the sum is even, make it odd by adding +1 and if the sum is odd do nothing

Write two decorators that would serve these two purposes and use them in the above function.