1. Write a program to find the length of the string without using inbuilt function (len)

2. Write a program to reverse a string without using any inbuilt functions.

3. Write a program to replace one string with another. e.g. "Hello World" replace "World" with "Universe".

4. How to convert a string to a list and vice-versa.

5. Covert the string "Hello welcome to Python" to a comma separated string.

6. Write a program to print alternate characters in a string.

7. Write a Program to print ascii values of the characters present in a string.

8. Write program to convert upper case to lower case and vice-versa without using inbuilt method.

9. Write program to swap two numbers without using 3rd variable.

10. Write program to merge two different lists.

11. Write program to read a random line in a file. (ex. 50, 65, 78th line)

12. Write program to read a random lines in a file. (ex. I want read all lines 10th to 15th line)

13 Program to print last "N" lines of a file.

14. Write a program to check if the given string is Palindrome or not without using reversed method.

15 Write a program to search for a character in a given string and return the corresponding index.

16 Write a program to get the below output

sentence = "hello world welcome to python programming hi there"

d = {'h': ['hello', 'hi'], 'w': ['world', 'welcome'], 't': ['to', 'there'], 'p': ['python', 'programming']
}

17 Write a to replace all the characters with - if the character occurs more than once in a string

18 write a decorator that returns only positive values of subtraction

19 How to get the count of number of instances of a class that is being created.

20 Write a function which takes a list of strings and integers.If the item is a string it should print as is and if the item is integer of float it should reverse it.

21 Write a class named Simple and it should have iteration capability.

22 Write a Custom class which can access the values of dictionaries using d['a'] and d.a

23 Write a python program to get the below output

sentence = "Hi How are you"

o/p should be "iH woH era uoy"

25 Write a lambda function to add two numbers (a, b)

26 What is the output of the following

sentence = "Hi How are you"

o/p should be "ouy era woH iH"

27 How to remove duplicates from the list without using inbuilt functions

```
>>> items = [1, 2, 3, 4, 1, 2, 3, 4, 5]
```

28 Find the longest word in the sentence

```
sentence = "Hello world. Welcome to Python"
```

29 write a program to reverse the values in the dictionary if the value is of type String

```
>>> d = {'a': 'hello', 'b': 100, 'c': 10.1, 'd': 'world'}
```

30 write a program to get 1234

```
t = ('1', '2', '3', '4')
```

31 How to get the elements that are in list b but not in list a

```
a = [1, 2, 3]
b = [1, 2, 3, 4]
```

32 A function takes variable number of positional arguments as input. How to check if the arguments that are passed are more than 5

33 Count the number of occurrences of "CRITICAL", "INFO" and "ERROR" lines in a log file.

```
# Assume Below is the contents of the log file

lines = """CRITICAL:Hello world
INFO: This is an info
ERROR: This is an error
CRITICAL: This is critical
CRITICAL:Hello world
INFO: This is an info
ERROR: This is an error
```

```
CRITICAL: This is critical

CRITICAL:Hello world

INFO: This is an info

ERROR: This is an error

CRITICAL: This is critical

CRITICAL:Hello world

INFO: This is an info

ERROR: This is an error

CRITICAL: This is critical"""
```

34 Write a function to reverse any iterable without using reverse function.

```
>>> a = [1, 2, 3, 4, 5]
```

35 Write a function to print the below output.

```
# func("TRACXN", 0)  # Should print RCN

# func("TRACXN", 1)  # Should print TAX
```

36 Sum all the numbers in the below string.

```
s = "Sony12India567Pvt2ltd"
```

37 Write a program to sum all the numbers in below string.

```
s = "Sony12India567Pvt2ltd" # eg.12+567+2
```

38 Print all the numbers in the below list

```
a = ['abc', '123', 'hello', '23']
```

39 Program to print the number of occurrences of characters in a String without using inbuilt functions.

```
>>> s = 'helloworld'
```

40 Program to print only the repeated characters and count of the same.

```
>>> s = 'helloworld'
```

41 Write a program to get alternate characters of a string in list format.

```
s = 'hello world welcome to python'
```

42 Write a program to get square of list of number's using lambda function .

```
>>> a = [1, 2, 3, 4, 5]
```

43 Write a function that accepts two strings and returns True if the two strings are anagrams of each other.

44 Write a program to iterate through list and build a new list, only if the items of the list has even number of characters.

```
>>> names = ['apple', 'yahoo', 'google', 'gmail', 'walmart', 'flipkart', 'facebook', 'amazon']
```

45 Write a program to iterate through list and build a new dictionary, only if the items of the list has even number of characters.

```
names = ['apple', 'yahoo', 'google', 'gmail', 'walmart', 'flipkart', 'facebook', 'amazon']
```

46 Write a program which squares the numbers in a list using map object

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

47 Count number of lines in a file without loading the file to the memory

48 Printing line and line no's

49 Write a Program to print the sum of entire list and sum of only internal list

```
l = [[1,2,3],[4,5,6],[7,8,9]]
```

50 Write a program to reverse the list as below

```
words = ["hi", "hello", "python"]
# o/p ['nohtyp', 'olleh', 'ih']
```

51 Write a program to update the tuple

t1 = (1, 2, 3, 4)

t2 = (100, 200, 300)

# o/p (1, 2, 3, 4, 100, 200, 300)

52 Write a program to replace value present in nested dictionary.

d = {'a': 100, 'b': {'m': 'man', 'n': 'nose', 'o': 'ox', 'c': 'cat'}}

# Replace "nose" with "net"

53 Write a program to count the number of white spaces in a file.

54 Grouping anagrams.

>>> words = ['eat', 'ate', 'tea', 'hello', 'silent', 'listen']

55 What is the difference between defaultdict and normal dictionary.

"""

Defaultdict

----

1. When each key is encountered for the first time, it will not be there in the mapping.

2. So an entry is automatically created with default value (an empty list in case of defaultdict of list and zero in case of defaultdict int).

3. When keys are encountered again, the look-up proceeds normally as like a normal dictionary.

4. So, in defaultdict, creation of key, initialisation will happen simultaneously.

Normal Dictionary

----

1. In case of normal dictionary, if the key does not exist, "KeyError" is raised.

2. In order to work on the value, first the key needs to be created and initialised.

"""

56 Explain property decorator in python.

```python

#

```

57 What is Mutable and Immutable datatypes.

```python

"""

1. Mutable datatypes are objects whose value can be changed after creation. e.g. list, dict, set, user defined classes.

2. Immutable datatypes are objects whose value can not be changed after creating. e.g. int, float, bool, tuple, namedtuple

"""

```

58 Explain get() method in dictionaries.

```python

"""

point =  {'a': 1, 'b': 2}

1. Values of dictionary can be accessed in two different ways. using square bracket syntax and the other one is using get() method.

2. When we try to access a key of a dictionary which does not exist using square bracket syntax (point['c']), "KeyError" exception is raised.

3. When we try to access a key of a dictionary which does not exist using get() method (point.get('c')), None is returned and no exception is raised.

4. We can pass a positional argument to get() method as custom message, so that get() method returns the custom message if the key does not exist.

        e.g. profile.get('c', 'Sorry the key does not exist')

"""

```

59 Write a list comprehension to get a list of even numbers from 1-50

60 Find the longest non-repeated substring in the below string

>>> s = "This is a Programming language and Programming is fun"

61 Write a program to find the duplicate elements in the list without using inbuilt functions

names = ['apple', 'google', 'apple', 'yahoo', 'google']

62 Write a program to count the number occurrences of each item in the list without using any inbuilt functions

names = ['apple', 'google', 'apple', 'yahoo', 'google', 'facebook', 'gmail', 'yahoo']

63 Write a function to check if the number is Prime

64 How to create a tuple using range function

65 Write a program to find the largest number in the list without using any inbuilt functions

>>> numbers = [10, 20, 30, 40, 50]

66 Write a method that returns the last digit of an integer. For example, the call of get_last_digit(3572) should return 2.

67 Write a program to find most common words in a given list.

words = [

'look', 'into', 'my', 'eyes', 'look', 'into', 'my', 'eyes',

'the', 'eyes', 'the', 'eyes', 'the', 'eyes', 'not', 'around', 'the', 'eyes', "don't", 'look', 'around', 'the',

'eyes', 'look', 'into','my', 'eyes', "you're", 'under'

]

68 Make a function named tail that takes a sequence (like a list, string, or tuple) and a number n and returns the last n elements from the given sequence, as a list.

69 Write function named is_perfect_square that accepts a number and returns True if it's a perfect square and False if it's not.

70 Write a program to get all the duplicate items and the number of times the item is repeated in the list.

```
>>> names = ['apple', 'google', 'apple', 'yahoo', 'yahoo', 'facebook', 'apple', 'gmail', 'gmail',
'gmail', 'gmail']
```

71 Write a program to count the number of occurrences of each word in a file.

72 Write a program to count the number of occurrences of vowels in a file.

73 Write a program to print all numeric values in a list

```
items = ['apple', 1.2, 'google', '12.6', 26, '100']
```

74 Triangle Patterns

```
*

* *

* * *

* * * *

* * * * *
```

```
    *

   * *

  * * *

 * * * *

* * * * *
```

```
    *

   * *

  * * *

 * * * *
```

```
* * * * *
```

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

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

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

# Number Pattern in triangle (Left Justified)

```
1
12
123
1234
```

```
12345

# Number Pattern in triangle (Right Justified)

    1
   12
  123
 1234
12345

# Number Pattern in triangle (Centre)

     1
    1 2
   1 2 3
  1 2 3 4
 1 2 3 4 5
```

75 Write a program count the occurrence of a particular word in the file

76 Write a program to map a product to a company and build a dictionary with company and list of products pair

```python
>>> all_products = ['iPhone', 'Mac', 'Gmail', 'Maps', 'iWatch', 'Windows',
                'iOS', 'Google Drive', 'One Drive']

>>> # Pre-defined products for different companies
>>> apple_products = {'iPhone', 'Mac', 'iWatch'}
>>> google_products = {'Gmail', 'Maps', 'Google Drive'}
>>> msft_products = {'Windows', 'One Drive'}
```

77 Write a program to rotate items of the list

```
>>> names = ["apple", "google", "yahoo", "gmail", "facebook", "flipkart", "amazon"]

>>> _rotate(names, 1)
>>> ['amazon', 'apple', 'google', 'yahoo', 'gmail', 'facebook', 'flipkart']
>>>
>>> _rotate(names, 2)
>>> ['flipkart', 'amazon', 'apple', 'google', 'yahoo', 'gmail', 'facebook']
```

78 Write a program to rotate characters in a string

```
>>> rotate_string("helloworld", 1)
>>> dhelloworl
>>> rotate_string("helloworld", 1)
>>> ldhellowor
```

```python
# Rotating words in a sentence
>>> sentence = "Hello world welcome to python"

>>> rotate(words, 1)
>>> python Hello world welcome to
>>> rotate(words, 2)
>>> to python Hello world welcome
```

79 Write a program to count the number of white spaces in a given string

```
>>> sentence = "Hello world welcome to Python Hi  How are you. Hi how are you"
```

80 Write a program to print only non-repeated characters in a string

>>> s = 'helloworld'

81 What is the difference between a list and a tuple

```python
"""
1. The main difference between a list and a tuple is that list's are mutable and tuples are immutable.

2. Python over allocates memory for lists. The reason for over allocation of memory is to support append operation.

Where as in tuples, memory is not over allocated, as tuples does not support append operation.

(Since tuples are immutable, it does not support append operation).

3. Tuples are more memory efficient than lists. (because in tuples no extra memory is allocated. It is fixed).

4. Tuples are negligibly faster than lists.
"""
```

82 Write a program to print all the consonants in a given string

>>> s = 'helloworld'

83 Write a program to count the number of commented lines in a text file

84 Write a program to check if the year is leap year or not

85 Liner Search

86 Difference between xrange and range

```python
"""
```

python2- xrange

python3- range

1. xrange does not stop, start and step attributes. But range object has start, stop and step attributes.

Python-3

r = range(0, 10)

r.start

r.stop

r.step

r = xrange(0, 10)

In Python-2 The above attributes are not supported.

2. range Object supports slicing! But xrange does not support slicing

3. range object has __contains__ method implemented. So it supports membership testing.

But xrange does not implement __contains__ method.

So Python will iterate over each and every item in the range xrange object until it finds a match.

(So if you are searching for a number in range is faster than xrange)

4. range will accept integer of any size. But xrange objects accepts integer size equivalent to C long!

"""

```

87 Write a program to count no of capital letters in a string

>>> sentence = "Hi How are You WelCome to PytHon"

88 Write a program to get the below output

*

* *
```

~~~
*
* * *
*
* * * *
*
* * * * *
~~~

89 Write a program to get the below output

~~~
>>> a = [1, 2, 3, 4, 5, 6, 7, 8, 9]


>>> [1, 2]
   [3, 4]
   [5, 6]
   [7, 8]
   [9]
```
~~~

90 Write a program to check if the elements in the second list is series of continuation of the items in the first list

a = [10, 12, 14, 16, 18]

b = [20, 22, 24, 26, 28]

a = [0, 5, 10, 15]

b = [20, 25, 30, 35, 40]

x = [10, 20, 30, 40]

y = [50, 40, 60, 70]

91 What is the difference between append() and extend() method in list

```python
"""
1. append() method appends one item at the end of the list.
2. extend() method appends all the items of the iterable to the end of the list.
3. Both append() and extend() method's mutates the existing list.

e.g.
>>> a = [1, 2, 3]
>>> b = (4, 5, 6)
>>> a.extend(b)
>>> a
[1, 2, 3, 4, 5, 6]

>>> c = {7, 8, 9}
>>> a.extend(c)
>>> a
[1, 2, 3, 4, 5, 6, 8, 9, 7]

>>> d = {'a': 1, 'b': 2, 'c': 3}
>>> a.extend(d)
>>> a
[1, 2, 3, 4, 5, 6, 8, 9, 7, 'a', 'b', 'c']

The list "a" is getting mutated each time when it is extended.
"""
```

92 Write a program to find the first repeating character in a string

>>> s = 'helloworld'

93 Write a program to find the index of nth occurrence of a sub-string in a string

```python
```

```
>>> sentence = "hello world welcome to python hello hi how are you hello there"
```

94 Write a program to print prime numbers from 1 to 50

95 Write a program to sort a list which has mix of both odd and even numbers, the sorted list should have odd numbers first and then even numbers in sorted order

```
>>> a = [3, 4, 1, 7, 2, 12, 8, 6, 9, 11]
```

96 Write a program to sort a list which has mix of both odd and even numbers, in the sorted list, odd numbers should be in ascending order and even numbers should be in descending order

```
>>> a = [3, 4, 1, 7, 2, 12, 8, 6, 9, 11]
```

97 Write a program to count the number of occurrences of non-special characters in a given string

```
>>> s = 'hello@world! welcome!!! Python$ hi how are you & where are you?'
```

98 Grouping Flowers and Animals in the below list

```
items = ['lotus-flower', 'lilly-flower', 'cat-animal', 'sunflower-flower', 'dog-animal']
```

99 Grouping files with same extensions

```
files = ['apple.txt', 'yahoo.pdf', 'gmail.pdf', 'google.txt', 'amazon.pdf', 'facebook.txt',
'flipkart.pdf']
```

100 Filter only those characters except digits

```
s = '@hello12world34welcome!123'
```

101 Count number of words in a sentence. ignore special characters.

```
>>> sentence = "Hi there! How are you:) How are you doing today!"
```

102 Grouping even and odd numbers

```
numbers = [1, 2, 3, 4, 5, 6, 7, 8, 9, 10]
```

103 Find all max numbers from the below list

```
>>> numbers = [1, 2, 3, 0, 4, 3, 2, 4, 2, 1, 0, 4]
```

104 Find all max length words from the below sentence

```
>>> sentence = "hello world hi apple you yahoo to you"
```

105 Find the range from the following string

```
>>> sentence = '0-0, 4-8, 20-20, 43-45'
>>> # Output Should be [0, 4, 5, 6, 7, 8, 20, 43, 44, 45, 46]
```

106 Can we override a static method in python?

```
class Parent:
    @staticmethod
    def demo():
        print('Running demo in Parent')


class Child(Parent):
    @staticmethod
    def demo():
        print('Running demo in Child')
```

107 Write a function which returns the sum of lengths of all the iterables

```
>>> total_length([1, 2, 3], (4, 5))
5
>>> total_length([1, 2, 3], (4, 5), ['apple', 'google', 'yahoo', 'gmail', 'flipkart'])
10
>>> total_length([1, 2, 3], (4, 5), ['apple', 'google', 'yahoo', 'gmail', 'flipkart'], {1, 2, 3})
```

```
13
>>> total_length([1, 2, 3], (4, 5), ['apple', 'google', 'yahoo', 'gmail', 'flipkart'], {1, 2, 3}, {'a': 1, 'b': 2})
15
```

108 Replace whitespaces with newline character in the below string

>>> sentence = "Hello world welcome to python"

109 Replace all vowels with "*"

>>> sentence = "hello world welcome to python"

110 Replace all occurances of "Java" with "Python" in a file

111 Maximum sum of 3 numbers and Minimum sum of 3 numbers

numbers = [10, 15, 20, 25, 30, 35, 40, 15, 15]

112 Write a program to get the output as below

# i/p is "python@#$%pool"

# o/p should be ['PYTHON', 'POOL']

113 Write a program to print all the number which are ending with 5

numbers = ['1', '12', '123', '12345', '125', '905', '55', '5', '95655', '55555']

114 Write a program to get the indexes of each item in the below list

names = ['apple', 'google', 'apple', 'yahoo', 'yahoo', 'google', 'gmail', 'gmail', 'gmail']

output should be - {'apple': [0, 2], 'google': [1, 5], 'yahoo': [3, 4], 'gmail': [6, 7, 8]}

115 Write a program to print "Bangalore" 10 times without using "for" loop

116 Write a program to print all the words which starts with letter 'h' in the given string

string = 'hello world hi hello universe how are you happy birthday'

117 Write a program to sum all even numbers in the given string

```
>>> sentence = 'hello 123 world 567 welcome to 9724 python'
```

118 Write a program to add each number in word1 to number in word2

```
# e.g. 1 + 5, 2 + 6, 3 + 7, 4 + 8, 5 + 9
>>> word1 = 'hello 1 2 3 4 5'
>>> word2 = 'world 5 6 7 8 9'
```

119 Write a program to filter out even and odd numbers in the given string

```
>>> sentence = 'hello 123 world 456 welcome to python498675634'
```

120 Write a program to print all the number which are starting with 8

```
>>> numbers = ['857', '987', '8', '120', '888888', '547', '7674', '89', '589', '388888', '2889']
```

121 Write a program to remove duplicates from the list without using set or empty list

```
>>> l1 = [1, 2, 3, 4, 1, 2, 3, 4, 3, 4, 4]
```

122 Print all the missing numbers from 1 to 10 in the below list

```
>>> numbers = [1, 3, 6, 8, 10]
```

123 Write a python program to get the below output

```
>>> l1 = [1, 2, 3]
>>> l2 = ['a', 'b', 'c']
>>> # o/p ['1a', '1b', '1c', '2a', '2b', '2c', '3a', '3b', '3c']
```

124 Write a python program to get the below output

```
>>> word = "AAAAaaccYYY"
>>> # o/p ['Y3', 'c2', 'a2', 'A4']
```

125 What is the output of the below function call

```
class Demo:
    def greet(self):
        print("hello world")

    def greet(self):
        print("hello universe")
```

126 In the list below, find all the number pairs which results in 10 either when added or subtracted

```
>>> a = [3, 5, -4, 8, 11, 1, -1, 6]
```

127 Write a decorator to prefix +91 to the original phone numbers

```
numbers = [1234567890, 123456790, 1234567890]
```

128 Write a program to get the below output

```
>>> d = {"a": 1, "b": 2, "c": 3, "d": 4, "e": 5}
>>> # o/p should be ['b', 'd']
```

129 Can we have multiple __init__ methods in a class

```python
class Point:
    def __init__(self, a, b):
        self.a = a
        self.b = b

    def __init__(self, a, b, c):
        self.a = a
        self.b = b
        self.c = c
```

130 Why python is Object Oriented

```python
        Python is object oriented because everyting in python is an object (defined as class)
```

131 What are .pyc files

```python
1. pyc files are python compiled.

2. Once .py file is compiled by python compiler, it generates .pyc file.

3. .pyc files contains byte code which is understandable by python virtual machine.

4. pyc files are generated when a python module is imported.

5. Python compiler will not compile a python module again and again unless there is a change in code.

6. This makes programs to run faster since byte code for a module is already generated.
```

132 Reverse a list without using any built-in functions and slicing

>>> a = [1, 2, 3, 4, 5]

133 Write a program to get the below output

>>> # i/p = "10.20.30.40"

>>> # o/p = "40.30.20.10"

134 What is the difference between while loop and for loop.

```python
The body of while loop gets executed until some condition is True.

Once the condition if False, the control comes out of the while loop.


The body of for loop get executed for some fixed number of iterations.
```

135 What are magic methods?

```python
```

Magic methods are special methods which starts and ends with double underscores.(they are also called as dunders)

Magic methods are internally called by python. We can customize the behaviour of an object or class using magic methods.

They are also called as protocols.

e.g. when you ask for the length of the list len(names) internally python will call __len__ method on list object.

e.g. when you check for membership "apple" in names python internally triggers __contains__("apple")

136 Swap two variables without using 3rd variable in python.

137 What is pylint.

```python

Pylint is a plugin or extension that checks for syntax errors.

Also, it tries to enforce coding standards according to PEP8 style guide.

It can give recommendations/suggestions/hints about types.

```

138 What is the output of the below program.

>>> [1, 2, 3, 4] * 2

139 What is the difference between the is and == operators

```python

"==" operator checks if both objects have same value.

"is" operator checks if identity or memory address of two objects are same.

```

140 What is "self" in class.

"self" is called as Instance or data.

Every instance method of a class has "self" as first argument.

During runtime, "self" will be replaced with object instance of the class.

e.g.

```
class Point:
   def __init__(self, a, b):
       self.a = a
       self.b = b


p1 = Point(1, 2)
p2 = Point(3, 4)
```

when you say, p1.a, "self" will be replaced with p1 and when you say p2.a, "self" will be replaced with p2.

Internally p1 and p2 will be pointing to a dictionary which is also called instance dictionary.

the instance dicitonary canbe accessed via __dict__ attribute.

e.g. p1.__dict__ , p2.__dict__

141 What is assert statement. What is the difference between assert and if/else statement.

1. An assert statement is used to valdate the actual result against expected result.

2. If the actual result does not match with the expected result, AssertionError is raised.

3. if/else is not used for validating the actual result against expected result.

4. if/else statement will not raise any exception.

142 What is the difference between a module, a package, and a library

1. A module is simply a Python file that’s intended to be imported into scripts or other modules.

It contains functions, classes, and global variables.

2. A package is a collection of modules that are grouped together inside a folder to provide consistent functionality. Packages can be imported just like modules. They usually have an __init__.py file in them.

3. A library is a collection of packages.

143 write a program to get the below output using while loop

```
1

12

123

1234
```

```
1

12

123

1234
```

144 write a program to get the below output

```
>>> items = ['$123.45', '$434.23', '$567.89']

>>> # o/p [123.45, 434.23, 567.89]
```

145 Generator function for Fibonacci series program

146 Write a program to print common character present in all the items of the below list

items = ["glory", "glass", "sight", "fight"]

147 Function should accept a list and if any number divisible by 3 then modify to 33 or else keep it as it is

148 write a program to print the below pattern

~~~
```python

1 2 3 *

1 2 * 4

1 * 3 4

* 2 3 4
~~~

149 write a program to map digits to its corresponding word

d = {   "0": "ZERO", "1": "ONE", "2": "TWO", "3": "THREE", "4": "FOUR",
        "5": "FIVE", "6": "SIX", "7": "SEVEN", "8": "EIGHT", "9": "NINE"
    }

150 validate if the list contains odd number at the begining (0th index) and even numbers there after from 1st till end of the list

numbers = [1, 2, 4, 8, 6, 12] ----> the function should return True

numbers = [1, 2, 4, 7, 6, 12] ----> the function should return False

numbers = [2, 2, 4, 8, 6, 12] ----> the function should return False

151 sort the list of names based on lastname or first character of the lastname

```python
names = ['steve jobs', 'bill gates', 'john doe', 'tim cook', 'laura turner', 'alex martin']
```

152 get all the pairs whose sum is 8

a = [1, 2, 3, 4, 5, 6, 7, 8, 9, 10]

153 write unique characters to the file

word = "aaabbbccc"

154 extract characters at even indexes of the string

items = (1, 2, 3, "bangalore", "mumbai")

155 Comparing two versions of the software

```python
from packaging.version import Version

# packaing is a standard library
```

~~~
a = "1.3.4"
b = "2.4.5"

v1 = Version(a)
v2 = Version(b)

v1 < v2     # returns True
v1 > v2     # returns False
v1 == v2    # returns False
```
~~~

156 Comparing two employee objects

~~~
```python
class Employee:
    def __init__(self, name, experience):
        self.name = name
        self.experience = experience

e1 = Employee('alex', 5)
e2 = Employee('brain', 1)
~~~

157 Replace characters at odd index by 'x'

~~~
>>> word = "example"
~~~

158 If the number is divisible by 2 it should print 'hi' if the no is divisible by 3 it should print 'hello' if the number is divisible by both 2 and 3 it should print 'bye'. using list comprehension

~~~
>>> numbers = [1, 2, 3, 4, 5, 6, 7, 8, 9, 10]
~~~

159 write a program to get the below output

```
# states = {'mysore':'karnataka','Bangalore':'karnataka','chennai':'TN','pune':'maharastra','coimbatore':'TN'}

# o/p {'karnataka': ['mysore', 'Bangalore'], 'TN': ['chennai', 'coimbatore'], 'maharastra': ['pune']}
```

160 write a program to get the below output

```
>>> l1 = ['m', 'na', 'i', 'pyt']

>>> l2 = ['y', 'me', 's', 'hon']

>>> # o/p ['my', 'name', 'is', 'python']
```

161 write a program to get the below output

```
>>> input={1:'a',2:'b',3:'c'}

>>> output = {'a': 1, 'b': 2, 'c': 3}
```

162 write a program to get the below output

```
>>> names = ['bangalore', 'chennai', 'mumbai', 'delhi']

>>> ['ore', 'nai', 'bai', 'lhi']
```

163 write a program to sort the given collection that contains uppercase, lowercase numeric and special character based on ASCII value

```
>>> items = ['a', 'b', 'C', 'D', 1, 8, '!']
```