

```

'''1. Write a program to find the length of iterable'''
## using len() inbuilt fun
# print(len('hello universe'))

## without using inbuilt func
# def length(iterable):
#     count = 0
#     for ele in iterable:
#         count += 1
#     return count
#
# print(length('python is a programming language'))

#-----
# def func():
#     return 'a', 'b', 10, True, 2.3, False, 'python'
#
# print(func()) #('a', 'b', 10, True, 2.3, False, 'python')

#####
'''2. write a program to reverse a string without any inbuilt methods'''
## Using slicing
# a = 'python'
# print(a[::-1])

## Using reversed
# a = 'python'
# print(reversed(a)) #<reversed object at 0x0000029B61446670>
# rev_ele = list(reversed(a))
# print(rev_ele) #['n', 'o', 'h', 't', 'y', 'p']
# print(''.join(rev_ele)) #nohtyp

## Alternate method
# a = 'python'
# for ele in range(len(a)-1, -1, -1):
#     print(a[ele], end='') #nohtyp

## Alternate method
# a = 'python'
# b = []
# for ele in range(len(a)-1, -1, -1):
#     b.append(a[ele])
# print(b)
# print(''.join(b))

# def reverse(string):
#     b = ''
#     for ele in string:
#         b = ele + b
#     print(b)
# reverse('spiderman')

# def reverse(list_):
#     b = []
#     for ele in list_:
#         b = [ele] + b
#     print(b)
# reverse(['apple', 'google', 'ajio', 'mynta', 'amazon', 'flipkart'])

#-----
'''3. wap to replace one string with another
str_='hello world', replace world with universe
'''

```

```

## replace()method
# str_ = 'hello world'
# print(str_.replace('world', 'universe'))          #hello universe

# str_ = 'hello world'
# import re
# print(re.sub('world', 'universe', str_))

#-----
'''wap to replace all the vowels in the string with *'''
# str_ = 'python is a programming language'
# result_str = ''
# for ele in str_:
#     if ele in 'aeiouAEIOU':
#         result_str += '*'
#     else:
#         result_str += ele
# print(result_str)

#-----
'''4. How to convert a string to a list and viceversa'''
# a = 'hello world'
# b = list(a)
# print(b)      ['h', 'e', 'l', 'l', 'o', ' ', 'w', 'o', 'r', 'l', 'd']

# print(a.split())      ['hello', 'world']
#-----
'''5. convert the string, 'Hello Welcome to Python' to a comma
seperated string'''
# str_ = 'Hello Welcome to Python'      #'Hello,Welcome,to,Python'
# print(str_.replace(' ', ','))      #Hello,Welcome,to,Python

## Alternate method
# str_ = 'Hello Welcome to Python'
# res = str_.split()
# print(','.join(res))      #Hello,Welcome,to,Python

#-----
'''6. wap to print alternate chracters in a string'''
# str_ = 'Hello Welcome to Python'
## Slicing
# print(str_[::2])      #HlWloet yhn

## Alternate method
# for ele in range(0, len(str_), 2):
#     print(str_[ele], end='')      #HlWloet yhn

#-----
'''7. wap to print the ASCII values of all the characters in the string'''
# str_ = 'hello world'
# for ele in str_:
#     print(f'The ASCII value of {ele} is {ord(ele)}')

#-----
'''8. wap to convert the uppercase alphabets to lowercase and viceversa
'''
## swapcase()
# str_ = 'HeLlO WeLcOmE To BEnGALuRu'
# print(str_.swapcase())      #hElLo wElCoMe tO beNgaUrU

## Alternate Solution
# str_ = 'HeLlO WeLcOmE To BEnGALuRu'
# res = ''
# for ele in str_:

```

```

#     if ele.isupper():
#         res += ele.lower()
#     elif ele.islower():
#         res += ele.upper()
#     else:
#         res += ele
# print(res)          #hElLo wElCoMe tO beNgalUrU

## Alternate
# str_ = 'HeLlO WeLcOmE To BEnGALuRu'
# res = ''
# for ele in str_:
#     if ord('a') <= ord(ele) <= ord('z'):
#         res += chr(ord(ele) - 32)
#     elif ord('A') <= ord(ele) <= ord('Z'):
#         res += chr(ord(ele) + 32)
#     else:
#         res += ele
# print(res)

#####
'''9. wap to swap two numbers without using 3rd variable'''
# num1 = 100
# num2 = 200
# num2, num1 = num1, num2
# print(num1)
# print(num2)

#-----
'''10. wap to merge two different lists'''
# list_1 = [10, '30', 20, 4.0]
# list_2 = ['hello', 'python', 'selenium']

#Using *
# list_3 = [*list_1, *list_2]
# print(list_3)          #[10, '30', 20, 4.0, 'hello', 'python', 'selenium']

#Using +
# print(list_1 + list_2)    #[10, '30', 20, 4.0, 'hello', 'python', 'selenium']

#Using extend method
# list_1.extend(list_2)
# print(list_1)    #10, '30', 20, 4.0, 'hello', 'python', 'selenium']

# from itertools import chain
# merged_list = chain(list_1, list_2)
# print(merged_list)    #<itertools.chain object at 0x0000024C8C6CD430>
# print(list(merged_list))    #[10, '30', 20, 4.0, 'hello', 'python', 'selenium']

#-----
## To merge dictionaries
# d1 = {'one':1, 'two':2, 'three':3}
# d2 = {1:'one', 2:'two', 3:'three'}
#
# # print(**d1, **d2)    #{'one': 1, 'two': 2, 'three': 3, 1: 'one', 2: 'two', 3: 'three'}
# from itertools import chain
# merged_dict = chain(d1, d2)
# # print(merged_dict)
# print(list(merged_dict))

#-----
'''11. wap to read a random line in a file'''
# path = r'C:\Users\QSP\PycharmProjects\PythonRevision\files\f1.txt'
#
# def random_line(num):

```

```

#     with open(path) as file:
#         for line_no, line in enumerate(file, start=1):
#             if num == line_no:
#                 print(f'{line_no} = {line}')
#
# random_line(9)

## Alternate solution
# from itertools import islice
# path = r'C:\Users\QSP\PycharmProjects\PythonRevision\files\f1.txt'
#
# def random_line(num):
#     with open(path) as file:
#         ## res = islice(file, line_num, line_num+1)
#         res = islice(file, num, num+1)
#         print(res) #<itertools.islice object at 0x000001E254E56090>
#         for ele in res:
#             print(ele)
#
# random_line(3)

#-----
'''11. wap to read a random lines in a file'''
# from itertools import islice
# path = r'C:\Users\QSP\PycharmProjects\PythonRevision\files\f1.txt'
#
# def random_line(start_val, end_val):
#     with open(path) as file:
#         ## res = islice(file, start, end)
#         res = islice(file, start_val, end_val)
#         print(res) #<itertools.islice object at 0x000001E254E56090>
#         for ele in res:
#             print(ele)
#
# random_line(3, 8)

#-----
'''wap to print last N lines of a file'''
# from itertools import islice
# path = r'C:\Users\QSP\PycharmProjects\PythonRevision\files\f1.txt'
#
# def last_n_lines(num):
#     with open(path) as file:
#         line_count = 0
#         for line in file:
#             line_count += 1
#         print(line_count)
#
#         file.seek(0)
#
#         res = islice(file, line_count-num, line_count)
#         # print(res)
#         print(list(res))
#
# last_n_lines(4)

## Alternate method
# from collections import deque
# path = r'C:\Users\QSP\PycharmProjects\PythonRevision\files\f1.txt'
#
# def last_n_lines(num):
#     with open(path) as file:
#         res = deque(file, num)
#         print(res)
#

```

```

# last_n_lines(3)

#-----
'''wap to check if the string is palindrome or not'''

# def palindrome(str_):
#     if str_ == str_[::-1]:
#         return f'{str_} is a palindrome'
#     else:
#         return f'{str_} is not a palindrome'
# print(palindrome('racecar'))

## Alternate
# def palindrome(str_):
#     rev_str = ''
#     for ele in str_:
#         rev_str = ele + rev_str
#
#     if str_ == rev_str:
#         return f'{str_} is a palindrome'
#     else:
#         return f'{str_} is not a palindrome'
#
# print(palindrome('malayalam'))
# print(palindrome('python'))
#-----
'''wap to search for a particular character in a given string
and return the corresponding index
'''
# def str_index(str_, element):
#     for index, item in enumerate(str_):
#         if element == item:
#             print(f'The index of {item} is {index}')
#
# str_index('welcome to python', 'e')

##Alternate
# def str_index(str_, n):
#     for ele in str_:
#         if str_.find(n) > -1:
#             return str_.find(n)
#
# print(str_index('python', 'p'))

##Alternate
# def string(str_, ele):
#     try:
#         print(str_.index(ele))
#     except ValueError:
#         print('element not found')
#
# string('python', 'l')

#-----
'''16 . wap to get the below output
str_ = 'python is a programming language and programs are fun
o/p --> {p:[python, programming, programs], i:[is], a:[a, and, are],
l:[language], f:[fun]}
'''
# str_ = 'python is a programming language and programs are fun'
# dict_ = {}
# split_str = str_.split()
# # print(split_str)
# # ['python', 'is', 'a', 'programming', 'language', 'and', 'programs', 'are', 'fun']
#

```

```

# for ele in split_str:
#     if ele[0] not in dict_:
#         dict_[ele[0]] = [ele]
#     else:
#         dict_[ele[0]] += [ele]
#
# print(dict_)

## Alternate Solution
## Using default dict

# str_ = 'python is a programming language and programs are fun'
#
# from collections import defaultdict
# def_dict = defaultdict(list)
# split_str = str_.split()
# for ele in split_str:
#     def_dict[ele[0]] += [ele]
#
# print(def_dict)

####-----
# a = 'she sells seashells on the seashore'
# d = {} # {s:8, h:4, e:7, ' ':5,...}
# for ele in a:
#     if ele in d:
#         d[ele] += 1
#     else:
#         d[ele] = 1
# print(d)

# from collections import defaultdict
# def_dict = defaultdict(int)
# for ele in a:
#     def_dict[ele] += 1
# print(def_dict)

#-----
'''17. wap to replace all the characters with * if the character
occurs more than once in a string'''
# str_ = 'hello welcome to Bengaluru'
##h*****w**c*m**t**B*nga**r*
# res = ''
# for ele in str_:
#     if str_.count(ele) > 1:
#         res += '*'
#     else:
#         res += ele
# print(res)

## Alternate solution using comprehension
# str_ = 'hello welcome to Bengaluru'
# result = ''.join(['*' if str_.count(ele)>1 else ele for ele in str_])
# print(result)

#####
'''18. write a decorator which will convert negative numbers to
positive'''
# def positive(func):
#     def wrapper(*args, **kwargs):
#         res = func(*args, **kwargs)
#         return abs(res)
#     return wrapper
#
# @positive
# sub = positive(sub)

```

```

# def sub(a, b):
#     return a - b
#
# print(sub(1232, 4565))

#-----
'''19. wap to get the count of number of instances of a class
that is being created'''
# class Sample:
#     count = 0
#
#     def __init__(self):
#         Sample.count += 1
#
# s1 = Sample()
# s2 = Sample()
# s3 = Sample()
# print(Sample.count)

#####
'''20. write a func which takes a list of strings and numbers.
If the item is a string, print it as it is, if it is int or
float, reverse it'''
# res_list = []
# def func(list_):
#     for ele in list_:
#         if isinstance(ele, str):
#             res_list.append(ele)
#         else:
#             res = str(ele)
#             res_list.append(res)
#
# func(['python', 24, 8.6, 'hello', 78, 9.8, 'welcome'])
# print(res_list)

#####
# l = ['apple', 'ajio']
# ## d = {apple:elppa, ajio:ajio}
# d = {}
# for ele in l:
#     if len(ele)%2 == 0:
#         d[ele] = ele
#     else:
#         d[ele] = ele[::-1]
# print(d)

#####
'''21. write a class named Sample and it should have iteration
capability'''
# class Sample:
#     def __init__(self, list_):
#         self.list_ = list_
#
#     def __iter__(self):
#         return iter(self.list_)
#
# s1 = Sample([1, 2, 3])
# for ele in s1:
#     print(ele)
#
# print(dir(Sample))

#-----
'''22. write a cuatom class which can access the values of
dictionaries using d['a'] and d.a '''

```

```

# class MyDict:
#
#     def __init__(self, dict_):
#         self.dict_ = dict_
#
#     def __getitem__(self, item):
#         return self.dict_[item]
#
#     # def __getattr__(self, item):
#     #     return self.dict_[item]
#
# d = MyDict({'a':1, 'b':2})
# print(d['a'])
# print(d['b'])

#-----
'''23. wap to get the below output
# s = 'hi how are you'          #ih woh era uoy'''
# s = 'hi how are you'          #ih woh era uoy
# words = s.split()
# print(words)                  #['hi', 'how', 'are', 'you']
# res_str = ''
# for ele in words:
#     res_str += ele[::-1] + ' '
#
# print(res_str)

#-----
'''24. wap to get the below output
s1 = 'hi how are you'          #ouy era woh ih'''
# s1 = 'hi how are you'          #ouy era woh ih
# res = ''
# for ele in s1:
#     res = ele + res
# print(res)

#-----
'''25. Write a lambda func to add two numbers'''
# res = lambda a, b : a + b
# print(res(6, 7))

#-----
'''26. What is the output of the following'''
# l1 = [1, 2, 3]
# l2 = [4, 5, 6]
# print([l1, l2])
# print((l1, l2))
#
# print([l1, l2])              #[[1, 2, 3], [4, 5, 6]]
# print((l1, l2))              #([1, 2, 3], [4, 5, 6])

#-----
'''27. wap to remove the duplicates from the list without using
inbuilt methods
'''
# items = [1, 2, 1, 3, 5, 4, 3, 6, 7, 4]  ##[1, 2, 3, 5, 4, 6, 7]
# result = set(items)
# print(result)  #{1, 2, 3, 4, 5, 6, 7}
# print(list(result))

##Alternate method
# items = [1, 2, 1, 3, 5, 4, 3, 6, 7, 4]  ##[1, 2, 3, 5, 4, 6, 7]
# unique_list = []
# for ele in items:
#     if ele not in unique_list:
#         unique_list.append(ele)

```



```

# print(unique_list)

# items = [1, 2, 1, 3, 5, 4, 3, 6, 7, 4] ##[1, 2, 3, 5, 4, 6, 7]
# unique_list = []
# for ele in items:
#     if items.count(ele) == 1:
#         unique_list.append(ele)
# print(unique_list)

##Using comprehension
# res = [ele for ele in items if ele in unique_list]
# print(res)

#-----
'''28. Find the longest word in the sentence'''
# s = 'python is a programming language'
# s_split = s.split()
# # print(s_split) #['python', 'is', 'a', 'programming', 'language']
#
# longest_word = ''
# for ele in s_split:
#     if len(ele) > len(longest_word):
#         longest_word = ele
#
# print(longest_word)

## Alternate method
# s = 'python is a programming language'
# longest = max(s.split(), key=len)
# print(longest)

'''Program to find the shortest element'''
# s = 'python is a programming language'
# shortest = min(s.split(), key=len)
# print(shortest)

## Alternate method
# s = 'python is a programming language'
# s_split = s.split()
# # print(s_split) #['python', 'is', 'a', 'programming', 'language']
# shortest_word = s_split[0]
# for ele in s_split:
#     if len(ele) < len(shortest_word):
#         shortest_word = ele
# print(shortest_word)

#-----
'''29. wap to reverse the values in a dict if the value is of type
string'''
# d = {'a':'hello', 'b':9.8, 'c':'world', 'd':100}
#d = {'a':'olleh', 'b':9.8, 'c':'dlrow', 'd':100}

# d1 = {}
# for key, value in d.items():
#     if isinstance(value, str):
#         d1[key] = value[::-1]
#     else:
#         d1[key] = value
# print(d1)

## Using Comprehension
# d2 = {key:value[::-1] if isinstance(value, str) else value for key, value in d.items()}
# print(d2)

```

```

#-----
'''30. wap to get 1234'''
# t = ('1', '2', '3', '4')          #1234
# print(''.join(t))

## Alternate solution
# t = ('1', '2', '3', '4')          #1234
# res = ''
# for ele in t:
#     res += ele
# print(res)

#-----
'''31. How to get the elements that are in list b, but not in
list a'''
# list_a = [1, 2, 3]
# list_b = [1, 2, 3, 4]
# set_a = set(list_a)
# set_b = set(list_b)
# print(set_b.difference(set_a))

## Alternate solution
# list_a = [1, 2, 3]
# list_b = [1, 2, 3, 4]
#
# for ele in list_b:
#     if ele not in list_a:
#         print(ele)

#-----
'''32. A function takes variable number of positional arguments
as input. How to check if the arguments that are passed are more
than 5
'''
# def func(*args):
#     if len(args) > 5:
#         print('The arguments are greater than 5')
#     else:
#         print('The arguments are lesser than 5')
#
# func('a', 'b', 10, 2, 3, 4, 'python', 'hai')
# func(1, 2)

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

# lines = '''CRITICAL:Hello world
# INFO: This is an info
# ERROR: this is an error
# CRITICAL: This is critical
# CRITICAL: hello
# INFO: hello info
# ERROR: This is error
# CRITICAL: This is critical
# CRITICAL: Critical issue
# INFO: hello
# ERROR: hello
# '''
# d = {}
# for line in lines.split('\n'):
#     split = line.split(':')
#     for ele in split:

```

```

#         if ele not in d:
#             d[ele] = 1
#         else:
#             d[ele] += 1
#
# print(d)

#-----
'''34. wap to reverse a iterable without using reversed func'''
# l = [1, 2, 3, 4, 5]
# rev_l = []
# for ele in range(len(l)-1, -1, -1):
#     rev_l.append(ele)
# print(rev_l)

#-----
'''35. Print the numbers in the below list'''
# l = ['abcd', '123', 'xyz', '456']
# for ele in l:
#     if ele.isnumeric():
#         print(ele)

#-----
'''36. write a func to print the below output
func('TRACXN', 0)  ---->Should print RCN
func('TRACXN', 1)  ---->Should print TAX
'''
# def func(string, num):
#     if num == 0:
#         print(string[1::2])
#     elif num == 1:
#         print(string[:2])
#     else:
#         print('Invalid Input')
#
# func('TRACXN', 0)      #RCN
# func('TRACXN', 1)      #TAX
# func('TRACXN', 8)

#-----
'''37. Sum of all the numbers in the below string'''
import re

# s = 'Sony12India567Pvt2ltd'
# total = 0
# res = re.findall(r'[\d]', s)
# print(res)  #['1', '2', '5', '6', '7', '2']
# for num in res:
#     total += int(num)
# print(total)

# s = 'Sony12India567Pvt2ltd'
# total = 0
# for ele in s:
#     if '0' <= ele <= '9':
#         total += int(ele)
# print(total)

#-----
'''38. wap to print sum of all numbers in below string'''
# s = 'Sony12India567Pvt25ltd'      #12 + 567 + 25
# total = 0
# res = re.findall(r'[\d]+', s)
# print(res)      #['12', '567', '25']
# for num in res:

```

```

#     total += int(num)
# print(total)

#-----
'''39. wap to print the number of occurrences of characters in
a string without using inbuilt methods'''
# s = 'hello world'
# d = {}
# for ele in s:
#     if ele in d:
#         d[ele] += 1
#     else:
#         d[ele] = 1
# print(d)

## Using defaultdict
# from collections import defaultdict
# s = 'hello world'
# def_dict = defaultdict(int)
# for ele in s:
#     def_dict[ele] += 1
# print(def_dict)

#-----
'''40. wap to print only the repeated characters and the count
of the repeated characters'''
# s = 'hello world'
# from collections import defaultdict
#
# def_dict = defaultdict(int)
# for ele in s:
#     def_dict[ele] += 1
# # print(def_dict)
#
# for key, value in def_dict.items():
#     if value > 1:
#         print(key, value)

#-----
'''41. wap to get alternate characetrs of a str in list format'''
# s = 'hello universe'
# res = s[::2]
# print(list(res))          #['h', 'l', 'o', 'u', 'i', 'e', 's']

## Using Comprehension
# s = 'hello universe'
# l = [ele for ele in s[::2]]
# print(l)                  #['h', 'l', 'o', 'u', 'i', 'e', 's']

#-----
'''42. wap to get the square of list of numbers'''
# l = [1, 3, 34, 32, 7, 65, 43, 10]
# def square(iterable):
#     square_list = []
#     for num in iterable:
#         square_list.append(num ** 2)
#     return square_list
#
# print(square(l))

## Using Lambda
# l = [1, 3, 34, 32, 7, 65, 43, 10]
# res = lambda num : num ** 2
# sq_list = [res(ele) for ele in l]
# print(sq_list)

```

```

#-----
'''43. write a func that accepts two strings and returns True
if the two strings are anagrams of each other '''
# def anagram(str1, str2):
#     if str1 == str2:
#         return False
#     a = sorted(str1)
#     b = sorted(str2)
#     if a == b:
#         return True
#     return False
#
# print(anagram('silent', 'listen'))
# print(anagram('racecar', 'racecar'))

#-----
'''44. write a program to iterate through list and build a
new list, only if the items in the list has even number of
characters'''
# list_ = ['mysore', 'bengaluru', 'bijapura', 'hubli', 'dharwad', 'badami']
# even_list = []
# for ele in list_:
#     if len(ele)%2 == 0:
#         even_list.append(ele)
#
# print(even_list)

#-----
'''45. write a program to iterate through list and build a
dict of even length ele as key and its len as value'''
# list_ = ['mysore', True, 'bengaluru', 10, 'bijapura', 'hubli', 'dharwad', 'badami']
# dict_ = {}
# for ele in list_:
#     if isinstance(ele, str):
#         if len(ele)%2 == 0:
#             dict_[ele] = len(ele)
#
# print(dict_)

#-----
'''46. wap which squares the numbers in a list using map object '''
# l = [1, 2, 3, 4, 5]
# ## map = map(func, iterable)
# res = map(lambda num : num ** 2, l) #[1, 4, 9, 16, 25]
# print(res) #<map object at 0x0000025A57CCD5B0>
# print(list(res))

## Using normal func
# l = [1, 2, 3, 4, 5]
# def square(num):
#     return num ** 2
#
# res = map(square, l)
# print(list(res))

#-----

# ele = 'bengaluru'
# res = lambda element : len(element) % 2 == 0
# print(res(ele))

# ele = 'bengaluru'
# res = lambda element : 'even length' if len(element) % 2 == 0 else 'odd length'
# print(res(ele))

# list_ = ['mysore', 'bengaluru', 'bijapura', 'hubli', 'dharwad', 'badami']

```

```

# res = map(lambda element : len(element) % 2 == 0, list_)
# print(res)
# print(list(res))    #[True, False, True, False, False, True]

# list_ = ['mysore', 'bengaluru', 'bijapura', 'hubli', 'dharwad', 'badami']
# res = filter(lambda element : len(element) % 2 == 0, list_)
# print(res)
# print(list(res))

#-----
'''47. wap to count the number of lines in a file without loading'
the file to the memory'''
# path = r'C:\Users\QSP\PycharmProjects\PythonRevision\files\f1.txt'
# with open(path) as file:
#     line_count = 0
#     for line in file:
#         line_count += 1
#     print(line_count)

#-----
'''48. Printing line and line numbers'''
# path = r'C:\Users\QSP\PycharmProjects\PythonRevision\files\f1.txt'
# with open(path) as file:
#     for line_no, line in enumerate(file, start=1):
#         print(line_no, '-', line)

#-----
'''49. wap to print the sum of entire list and sum of internal list'''
# l = [[1, 2, 3], [4, 5, 6], [7, 8, 9]]
## [6, 15, 24]
## 45
# sum_list = []
# for ele in l:
#     sum_list.append(sum(ele))
# print(sum_list)
# print(sum(sum_list))

# l1 = []
# total = 0
# for ele in l:
#     for item in ele:
#         total += item
#
# print(total)

#-----
'''50. wap to reverse the list as below'''
# words = ['hi', 'hello', 'python']
# op --> ['nohtyp', 'olleh', 'ih']
# rev_list = []
# for ele in words:
#     rev_list.append(ele[::-1])
# print(rev_list[::-1])

#-----
'''51. wap to merge the tuples'''
# t1 = (1, 2, 3, 4)
# t2 = (100, 200, 300)
# print(t1 + t2) #(1, 2, 3, 4, 100, 200, 300)
# print((*t1, *t2)) #(1, 2, 3, 4, 100, 200, 300)

#-----
'''52. wap to replace the value present in nested dictionary'''
# d = {'a':100, 'b':{'m':'man', 'n':'nose', 'c':'cat'}}

```

```

# ## Replace nose with net
# for key, value in d.items():
#     if isinstance(value, dict):
#         d[key]['n'] = 'net'
# print(d)    #{'a': 100, 'b': {'m': 'man', 'n': 'net', 'c': 'cat'}}

#-----
'''53. write a list comprehension to get list of even numbers
from 1-50'''
# even_ = [ele for ele in range(1, 51) if ele%2==0]
# print(even_)

# odd_ = [ele for ele in range(1, 51) if ele%2 != 0]
# print(odd_)

#-----
'''54. wap to count number of white spaces in a file'''
# import re
# path = r'C:\Users\QSP\PycharmProjects\PythonRevision\files\f1.txt'
# count = 0
# with open(path) as file:
#     for line in file:
#         match = re.findall(r'\s', line)
#         if match:
#             count += len(match)
#
# print(count)

#-----
'''55. difference between default dict and normal dict'''
...
default dict :
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
3. When keys are encountered again, the look-up proceeds normally
like a normal dictionary
4. So, in default dict, creation of the key, initialisation will
happen simultaneously

Normal dict : In case of normal dict, if the key doesnot exist,
'KEY ERROR' is raised
2. So, we first create the keys and initialise the values
...

#-----
'''56. Property decorator'''

#-----
'''57. Mutable and immutable datatypes'''

#-----
'''58. Explain get() method in dict'''
# dict_ = {'India':'Delhi', 'Srilanka':'Colombo', 'Pakistan':'Islamabad'}
# print(dict_['India'])    #Delhi
# print(dict_['USA'])
## In the above method, if the key is not found, it will throw
## KeyError.

# print(dict_.get('India'))
# print(dict_.get('USA'))    #None
# print(dict_.get('USA', 'key not found'))
## In case of get method, if the key is not found, get() method will
## not give any error, instead it will give None. Incase, we want
## any other value, we can pass the value as a second parameter

```

```

# print(dict_)

'''setdefault'''
# dict_ = {'India':'Delhi', 'Srilanka':'Colombo', 'Pakistan':'Islamabad'}
# print(dict_.setdefault('India'))
# print(dict_.setdefault('USA', 'Washington DC'))
# print(dict_)
#-----
'''60. Find the longest non-repeated substring in the below string'''
# s = 'python is a programming language and programming is very easy'
# words = s.split()
# ele = words[0]
# for word in words:
#     if words.count(word) == 1:
#         if len(word) > len(ele):
#             ele = word
# print(ele, len(ele))

## Alternate
# s = 'python is a programming language and programming is very easy'
# dict_ = {}
# words = s.split()
# for ele in words:
#     if words.count(ele) == 1:
#         dict_[ele] = len(ele)
# # print(dict_)
#
# res = sorted(dict_.items(), key=lambda item : item[-1])
# print(res[-1])

#-----
'''61. write a program to find the duplicate elements in the list
without using inbuilt functions'''
# list_ = ['apple', 'google', 'apple', 'gmail', 'google', 'kia']
# s1 = set(list_)
# # print(s1)
#
# for item in s1:
#     count = 0
#     for ele in list_:
#         if item == ele:
#             count += 1
#     if count > 1:
#         print(item)
#-----
'''62. wap to count the number of occurrences of each item in the
list without using inbuilt functions'''
# list1 = ['duster', 'seltos', 'kiger', 'ritz', 'kiger', 'duster']
# d = {}
# for ele in list1:
#     if ele in d:
#         d[ele] += 1
#     else:
#         d[ele] = 1
#
# print(d)
#-----
'''63. write a func to check if the number is prime'''
# def is_prime(num):
#     for i in range(2, num):
#         for j in range(2, i):
#             if i % j == 0:
#                 break
#     else:

```



```

#         print('prime')
# is_prime(7)

# def is_prime(num):
#     return not any(num % ele == 0 for ele in range(2, num))
#
# print(is_prime(9))
# print(is_prime(7))
# print(is_prime(17))
#-----
'''64. create a tuple using range'''
# print(range(1, 10))      #range(1, 10)
# print(tuple(range(1, 10)))  #(1, 2, 3, 4, 5, 6, 7, 8, 9)

#-----
'''65. wap to find the largest number in the list without using
any inbuilt func'''
# l = [1, 4, 10, 32, 8, 4, 2]
# large = l[0]
# for i in l:
#     if i > large:
#         large = i
# print(large)
#-----
'''66. write a func that returns the last digit of an integer.
For Eg. if user input is 3716, output should be 6.
'''
# def is_last_digit(num):
#     if num % 10 != 0:
#         temp = num % 10
#         print(temp)
#
# is_last_digit(4568709)

## Alternate
# def is_last_digit(num):
#     temp = str(num)
#     print(int(temp[-1]))
#
# is_last_digit(4568709)
#-----
'''67. wap to find the most common words in 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'
# ]
# d = {}
# for word in words:
#     if word in d:
#         d[word] += 1
#     else:
#         d[word] = 1
# print(sorted(d.items(), key=lambda item: item[-1])[-1])

# from collections import Counter
# c = Counter(words)
# print(c)      #Counter({'eyes': 7, 'the': 5, 'look': 3, 'into': 2, 'my': 2, 'around': 2,
# 'not': 1, "don't": 1})
# print(c.most_common())  #[('eyes', 7), ('the', 5), ('look', 3), ('into', 2), ('my', 2),
# ('around', 2), ('not', 1), ("don't", 1)]
# print(c.most_common()[0])

#-----
'''68. Make a func named tail that takes the sequence

```

```

(like list, str, tuple) and a number, and returns the last n
elements from the given sequence as a list'''
# def tail(iterable, n):
#     if not isinstance(n, int):
#         return 'Only integers'
#     if n < 0:
#         return []
#     return list(iterable[-n:])
#
# print(tail('programming', -4))
#-----
'''69. write a func that accepts a number and returns True
if its a perfect square, else False
'''
# from math import sqrt
# def is_perfect_square(num):
#     return num == sqrt(num)**2
#
# print(is_perfect_square(8))      #False
# print(is_perfect_square(9))      #True
#-----
'''70. wap 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']
# words = set(names)
# for ele in words:
#     res = names.count(ele)
#     if res > 1:
#         print(ele)
#-----
'''71. wap to count the number of occurances of each word in file'''
# from collections import defaultdict
# path = r'C:\Users\QSP\PycharmProjects\PythonRevision\files\sample_1.txt'
# with open(path) as file:
#     d = defaultdict(int)
#     for line in file:
#         if line.strip():
#             res = line.split()
#             for ele in res:
#                 d[ele] += 1
#
#     print(d)
#-----
#####
# def even(start, end):
#     for i in range(start, end):
#         if i % 2 == 0:
#             yield i
#
# res = even(1, 11)
# print(res)
# print(list(res))
#-----
'''72. wap to count the number of vowels'''
# path = r'C:\Users\QSP\PycharmProjects\PythonRevision\files\sample_1.txt'
# with open(path) as file:
#     for line in file:
#         if line.strip():
#             vowel_count = 0
#             for ele in line:
#                 if ele in 'aeiouAEIOU':
#                     vowel_count += 1

```

```

#             print(f'{line.strip()} has {vowel_count} vowels')

#-----
'''73. wap to print all numeric values in a list'''
# l = ['apple', 1.2, 'google', '26', 100, 78]
# for ele in l:
#     if isinstance(ele, (int, float)):
#         print(ele)

#-----
'''Traingle patterns'''
'''
*
* *
* * *
* * * *
'''
# for ele in range(1, 5):
#     print(ele * '*' )

#-----
'''
* * * *
* * *
* *
*
'''
# for i in range(4, 0, -1):
#     print(i * '*' )

#-----
## Right justified triangle
'''
      *
     * *
    * * *
   * * * *
'''
# for i in range(1, 5):
#     print(f'{"* " * i:>8}')
```

```

#-----
'''
* * * *
* * *
* *
*
'''
# for i in range(4, 0, -1):
#     print(f'{"* " * i:>8}')
```

```

#-----
'''
      *
     * *
    * * *
   * * * *
'''
# for i in range(1, 5):
#     print(f'{"* " * i:^8}')
```

```

#-----
'''
* * * *
* * *
'''
```

```

    * *
    *
'''
# for i in range(4, 0, -1):
#     print(f'{"* " * i:^8}')

#-----
'''
*
*
* *
*
* * *
*
* * * *
*
'''
# for ele in range(1, 5):
#     print(ele * '*'')
#     print('*')

#-----
'''
1
1 2
1 2 3
1 2 3 4
1 2 3 4 5
'''
# pat = ""
# for i in range(1, 6):
#     pat += str(i) + ' '      #pat = pat + str(i)
#     print(pat)

#-----
'''
        1
       1 2
      1 2 3
     1 2 3 4
    1 2 3 4 5
'''
# pat = ''
# for i in range(1, 6):
#     pat += str(i) + ' '
#     print(f'{pat :>10}')

#-----
'''
        1
       1 2
      1 2 3
     1 2 3 4
    1 2 3 4 5
'''
# pat = ''
# for i in range(1, 6):
#     pat += str(i) + ' '
#     print(f'{pat :^10}')

#-----
'''
a
a b
a b c

```

```

a b c d
'''
# pat = ""
# for ele in range(ord('a'), ord('e')): #for ele in range(97, 101)
#     pat += chr(ele) + ' '
#     print(pat)

#-----
'''75. wap that counts the occurrences of a particular word in file'''
# path = r'C:\Users\QSP\PycharmProjects\PythonRevision\files\sample_1.txt'

# def _count(word):
#     with open(path) as f1:
#         count = 0
#         for line in f1:
#             words = line.split()
#             for i in words:
#                 if i == word:
#                     count += 1
#         print(count)
# _count('hello')

#-----
'''76. wap to map a product to a company and build a dict with
company and list of products pair'''
# all_products = ['iPhone', 'mac', 'gmail', 'maps', 'iWatch',
#                 'windows', 'google drive', 'one drive']
#
# apple_products = ['iphone', 'mac', 'iwatch']
# google_products = ['gmail', 'maps', 'googledrive']
# msft_products = ['windows', 'onedrive']
#
# from collections import defaultdict
# def_dict = defaultdict(list)
#
# for ele in all_products:
#     if ele in apple_products:
#         def_dict['apple'] += [ele]
#     elif ele in google_products:
#         def_dict['google'] += [ele]
#     elif ele in msft_products:
#         def_dict['msft'] += [ele]
#
# print(def_dict)

#-----
'''77. wap to rotate the items of the list'''
# names = ['apple', 'google', 'gmail', 'yahoo', 'microsoft']
# def _rotate(iterable, n):
#     for i in range(0, n):
#         res = iterable.pop()
#         iterable.insert(0, res)
#     print(iterable)
#
# _rotate(names, 2)
# _rotate(names, 3)

#-----
'''78. wap to rotate characters in a string'''
# s = 'hello world'
# print(s[-3:] + s[:-3])

#-----
'''79. wap to count the number of white spaces in a given string'''

```

```

# s = 'hello world welcome to python'
# import re
# print(len(re.findall(r'\s', s)))

#-----
'''80. wap to print only non repeated characters in a string'''
# s = 'hello world'
# for ele in s:
#     if s.count(ele) == 1:
#         print(ele)

#-----
'''81. what is the difference between list and tuple'''

#-----
'''82. wap to print all the consonants in a given string'''
# s = 'hello world'
# res = [ele for ele in s if ele not in 'aeiouAEIOU ']
# print(res)

#-----
'''83. wap to check if the year is leap year or not'''
# import calendar
# print(calendar.isleap(2020))
# print(calendar.isleap(2023))

#-----
'''85. linear search'''
# a = range(1, 20)
# def func(num):
#     if num in a:
#         return True
#     else:
#         return False
#
# print(func(6))

#-----
'''86. wap to count the number of commented lines in the file'''
# path = r'C:\Users\QSP\PycharmProjects\PythonRevision\files\wednesday_apr_19.py'
# with open(path) as file:
#     count = 0
#     for line in file:
#         if line.strip():
#             if line.startswith('#'):
#                 count += 1
#     print(count)

#-----
'''87. wap to count no of capital letters in the string'''
# str_ = 'PyThoN 123 IS @ PrOgRAmmING 567 LAnGuagE !'
# count = 0
# for ele in str_:
#     if 'A' <= ele <= 'Z':
#         count += 1
#
# print(count)

#-----
'''87. wap to count no of capital letters in the string'''
# str_ = 'PyThoN 123 IS @ PrOgRAmmING 567 LAnGuagE !'
# spe_char_count = 0
# alpha_count = 0
# num_count = 0
# for ele in str_:
#     if ele.isalpha():

```

```
#         alpha_count += 1
#     elif ele.isdigit():
#         num_count += 1
#     else:
#         spe_char_count += 1
#
# print(spe_char_count)
# print(num_count)
# print(alpha_count)
```

```
#-----
'''88. difference between xrange and range
python2 - xrange
python3 - range
```

1. xrange donot have start, stop and step attributes
But range object has start, stop and step attributes
 2. range object supports slicing whereas xrange donot support slicing
 3. range object has __contains__, so it supports membership operator
But xrange doesnot implement __contains__ method
- ```
...
```

```
#-----
'''89. wap to get the below output'''
a = [1, 2 ,3, 4, 5, 6, 7, 8, 9]
'''op ---> [1, 2], [3, 4], [5, 6], [7, 8], [9]'''
for i in range(0, len(a), 2):
print(a[i:i+2])
```

```
#-----
'''90. wap to check if the elements in the second list
is series of continuation of the items in the first list'''
l1 = [2, 4, 6, 8, 10]
l2 = [12, 14, 16, 18, 20]
#
diff = l1[1] - l1[0]
l = [*l1, *l2] #[2, 4, 6, 8, 10, 12, 14, 16, 18, 20]
#
def continuation():
for ele in range(0, len(l)):
if l[ele + 1] == l[ele] + diff:
return True
else:
return False
#
print(continuation())
```

```
#-----
'''91. what is the difference between append method and extend
method in list'''
...
```

append method adds one item at the end of the list  
 extend method adda all the items of the iterable at the end  
 of the list  
 Both append and extend will add the elements to the end of the list

```
...
```

```
l = [1, 2, 3]
l.append('hai')
print(l)
#
l.extend('100')
print(l)
```

```

#-----
'''92. wap to find the first repeating character in a string'''
s = 'hai hello universe how are you'
for ele in s:
if s.count(ele) > 1:
print(ele)
break

#-----
'''93. wap to find the index of nth occurrence of a substring
in a string'''
import re
s = 'she sells seashells on the seashore'
def n_occurance(iterable, pat, n):
count = 0
res = re.finditer(pat, iterable)
for ele in res:
count += 1
if count == n:
print(ele)
#
n_occurance(s, 'e', 7)

#-----
'''write a func to check if the numb is prime or not'''
def prime(n):
for i in range(2, n):
if n % i == 0:
print(f'{n} is not prime')
break
else:
print(f'{n} is a prime number')
prime(29)

#-----
'''94. wap to find prime numbers in the range 2-50'''
def prime(n):
for i in range(2, n):
for j in range(2, i):
if i % j == 0:
break
else:
print(i)
#
prime(50)

#-----
'''95. wap to sort a list which has a mix of even and odd numbers
, the sorted list should contain odd numbers first and then even
numbers in a sorted order'''
num_list = [2, 17, 31, 32, 47, 97, 98, 60, 55, 46, 49]
#
even_list = [ele for ele in num_list if ele % 2 == 0]
odd_list = [ele for ele in num_list if ele % 2 != 0]
#
even_list.sort()
odd_list.sort()
#
print([*odd_list, *even_list])

#-----
'''96.wap to sort a list which has a mix of even and odd numbers
, the sorted list should contain odd numbers first and then even
numbers in a descending order'''
num_list = [2, 17, 31, 32, 47, 97, 98, 60, 55, 46, 49]

```



```

#
even_list = [ele for ele in num_list if ele % 2 == 0]
odd_list = [ele for ele in num_list if ele % 2 != 0]
#
even_list.sort(reverse=True)
odd_list.sort()
#
print(*odd_list, *even_list)

#-----
'''97. wap to count the number of occurrences of non-special
characters in the given string'''
s = '123hello @ world ! % ha! Un!verse4567 *()'
count = 0
for ele in s:
if ele.isalnum():
count += 1
#
print(count)

alternate
import re
s = '123hello @ world ! % ha! Un!verse4567 *()'
res = re.findall(r'\w', s)
print(len(res))

#-----
s = '123hello @ world ! % ha! Un!verse4567 *()'
from collections import defaultdict
def_dict = defaultdict(int)
#
for ele in s:
if ele.isalnum():
def_dict[ele] += 1
print(def_dict)

#-----
'''98. Grouping Flowers and animals in the below list'''
items = ['lotus-flower', 'lilly-flower', 'dog-animal',
'cat-animal', 'sunflower-flower']
from collections import defaultdict
def_dict = defaultdict(list)
for item in items:
value, key = item.split('-')
def_dict[key] += [value]
print(def_dict)

#-----
'''99. Grouping files with same extensions'''
files = ['apple.txt', 'yahoo.pdf', 'gmail.pdf', 'google.txt',
'amazon.pdf', 'facebook.txt', 'flipkart.pdf']
from collections import defaultdict
def_dict = defaultdict(list)
for item in files:
f_name, ext = item.split('.')
def_dict[ext] += [f_name]
print(def_dict)

#-----
'''100. Filter only those characters except digits'''
s = '123@hello4567 world 789 universe !!'
non_digits = []
for ele in s:
if not ele.isdigit():
non_digits.append(ele)

```

```

#
print(''.join(non_digits))

Alternate
s = '123@hello4567 world 789 universe !!'
import re
res = re.findall(r'^\d+', s)
print(''.join(res))

#-----
'''101. Count the number of words in a sentence, ignore
special characters'''
sentence = 'Hi there! How are you:) How are you doing today!'
import re
res = re.findall(r'[\w]+', sentence)
print(res) #['Hi', 'there', 'How', 'are', 'you', 'How', 'are', 'you', 'doing', 'today']
#
from collections import Counter
c = Counter(res)
print(c) #Counter({'How': 2, 'are': 2, 'you': 2, 'Hi': 1, 'there': 1, 'doing': 1,
'today': 1})

#-----
'''102. Grouping even and odd numbers'''
numbers = [1, 2, 3, 4, 5, 6, 7, 8, 9, 10]
o/p ---> {1:[1, 3, 5, 7, 9], 0:[2, 4, 6, 8, 10]}
#
from collections import defaultdict
def_dict = defaultdict(list)
for num in numbers:
if num % 2 == 0:
def_dict[0] += [num]
else:
def_dict[1] += [num]
#
print(def_dict)

#-----
'''103. Find all the max numbers from the below list'''
l = [1, 2, 2, 5, 6, 7, 8, 9, 4, 9, 9, 8, 9]
max_ = max(l)
print(max_)
print(l.count(max_))

Alternate
l = [1, 2, 2, 5, 6, 7, 8, 9, 4, 9, 9, 8, 9]
max_ = max(l)
max_list = []
for ele in l:
if ele == max_:
max_list.append(ele)
#
print(max_list) # [9, 9, 9, 9]

#-----
'''104. Find all max length words from the below sentence'''
sentence = 'hello world hi apple hai yahoo'
res = sentence.split()
print(res) # ['hello', 'world', 'hi', 'apple', 'hai', 'yahoo']
#
max_len = max(res)
for ele in res:
if len(ele) == len(max_len):
print(ele)

```

```

#-----
'''105. Find the range from the following string'''
s = '0-0,4-8,20-20,43-45'
##[0, 4, 5, 6, 7, 8, 20, 43, 44, 45]
res = s.split(',')
print(res) #['0-0', '4-8', '20-20', '43-45']
#
range_list = []
for ele in res:
start, end = ele.split('-')
for num in range(int(start), int(end)+1):
range_list.append(num)
#
print(range_list)
#-----
'''106. Can we override static method in Python'''
class Demo:
#
@staticmethod
def display():
print('Demo class display executing')
#
class Spam(Demo):
#
@staticmethod
def display():
print('Spam class display executing')
#
spam_obj = Spam()
spam_obj.display()

#-----
'''107. write a func which returns the sum of lengths of all
the iterables'''
def total_length(*iterables):
total = 0
for iterable in iterables:
total += len(iterable)
return total
#
print(total_length('hai', [1, 2, 3, 4], (2, 3, 4), {1:1, 2:2}))

#-----
'''108. Replace whitespaces with newline character in the below
string'''
sentence = 'Hello world welcome to python'
print(sentence.replace(' ', '\n'))

#Alternate
import re
sentence = 'Hello world welcome to python'
res = re.sub(r'\s', '\n', sentence)
print(res)
#-----
'''109. Replace all the vowels with "*"'''
sentence = 'Hello world welcome to python'
import re
res = re.sub(r'[aeiou]', '*', sentence)
print(res)

##Alternate
sentence = 'Hello world welcome to python'
result = ''
for ele in sentence:
if ele in 'aeiouAEIOU':

```

```

result += '*'
else:
result += ele
print(result)
#-----
'''110. Replace all the occurrences of Java with Python
in a file'''
path = r'C:\Users\QSP\PycharmProjects\PythonRevision\files\sample_1.txt'
with open(path) as file:
for line in file:
res = re.sub(r'java', 'python', line)
print(res)

#-----
'''111. wap to find sum of max 3 numbers and min 3 numbers'''
l = [9, 1, 4, 2, 3, 8, 6, 7, 5]
#
sorted_list = sorted(l)
print(sorted_list) #[1, 2, 3, 4, 5, 6, 7, 8, 9]
#
min_sum = sum(sorted_list[:3])
print(min_sum)
#
max_sum = sum(sorted_list[-3:])
print(max_sum)

#-----
'''112. wap to get the below output
input --> 'python@#$$pool'
output --> ['PYTHON', 'POOL']
'''
s = 'python@#$$pool'
import re
res = re.findall(r'\w+', s)
print(res)
result = [ele.upper() for ele in res]
print(result)

#-----
'''113. wap to print all the numbers which are ending with 5'''
numbers = ['1', '12', '123', '12345', '125', '155', '555', '666']
for num in numbers:
if num.endswith('5'):
print(num)

Alternate
numbers = ['1', '12', '123', '12345', '125', '155', '555', '666']
for number in numbers:
res = re.findall(r'\d+5$', number)
if res:
print(res)

#-----
'''114. wap to get the indexes of each item in the below list'''
names = ['apple', 'google', 'apple', 'gmail', 'yahoo', 'gmail',
'microsoft', 'yahoo', 'flipkart']
#
from collections import defaultdict
def_dict = defaultdict(list)
#
for index, item in enumerate(names):
def_dict[item] += [index]
print(def_dict)

#-----

```

```

'''115. wap to print "Bengaluru" for 10 times without using loops '''
print('Bengaluru\n' * 10)

#-----
'''116. wap to print all the words which starts with letter 'h'
in the given string '''
s = 'hello world hi universe how are you happy birthday'
res = re.findall(r'\bh[\w]+', s)
print(res)

Alternate
s = 'hello world hi universe how are you happy birthday'
l = []
res = s.split()
for ele in res:
if ele.startswith('h'):
l.append(ele)
#
print(l)

#-----
'''117. wap to find the sum of all even numbers in the given string'''
s = 'hello 123 world 567 welcome to 9724 python'
res = re.findall(r'\d', s)
print(res) #['1', '2', '3', '5', '6', '7', '9', '7', '2', '4']
#
sum_ = 0
for ele in res:
if int(ele) % 2 == 0:
sum_ += int(ele)
#
print(sum_)

#-----
'''118. wap to add each num in word1 to number in word2'''
word1 = 'hello 1 2 3 4 5'
word2 = 'world 5 6 7 8 9'
output --> 1+5, 2+6, 3 +7, 4+8, 5+9
#
num1 = re.findall(r'\d', word1)
num2 = re.findall(r'\d', word2)
print(num1) #['1', '2', '3', '4', '5']
print(num2) #['5', '6', '7', '8', '9']
#
for i, j in zip(num1, num2):
print(int(i) + int(j))

#-----
'''119. wap to filterout even and odd numbers in the given str'''
s = 'hello 123 world 567 welcome to 9724 python685645'
res = re.findall(r'\d', s)
print(res) #['1', '2', '3', '5', '6', '7', '9', '7', '2', '4', '6', '8', '5', '6', '4', '5']
#
even_list = [int(ele) for ele in res if int(ele)%2==0]
odd_list = [int(ele) for ele in res if int(ele)%2!=0]
#
print(even_list)
print(odd_list)

#-----
'''120. wap to print all the numbers starting with 8'''
numbers = ['8', '87', '67', '890', '45', '888', '108', '828']
for num in numbers:
if num.startswith('8'):
print(num)

```

```

Alternate
numbers = ['8', '87', '67', '890', '45', '888', '108', '828']
for number in numbers:
res = re.findall(r'^8\d+', number)
if res:
print(res)

#-----
'''121. wap to remove duplicates from the list without using
empty list or converting to set'''
l1 = [1, 2, 3, 4, 1, 2, 3, 4, 3, 4, 4, 4, 3]
l2 = l1[:]
for ele in l2:
if l1.count(ele)>1:
l1.remove(ele)
print(l1)

#-----
'''122. Print all the missing numbers from 1-10 in the below list'''
l = [1, 3, 6, 8, 10]
l1 = []
for num in range(1, 11):
if num not in l:
l1.append(num)
print(l1)

#-----
'''123. wap to get the below output'''
l1 = [1, 2, 3]
l2 = ['a', 'b', 'c']
1a, 2a, 3a, 1b, 2b, 3b, 1c, 2c, 3c
for i in l1:
for j in l2:
print(str(i)+j)

#-----
'''124. wap to get the below output
o/p ---> [Y3, c2, a2, A4]'''

words = 'AAAAaaccYYY'
unique_words= set(words)
print(unique_words)
#
list_ = []
for ele in unique_words:
temp = words.count(ele)
list_.append(ele+str(temp))
print(list_)

#-----
'''125. what is the output'''
class Demo:
def greet(self):
print('hello world')
#
def greet(self):
print('hello universe')
#
d = Demo()
d.greet() #hello universe

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

```

```

l = [3, 5, -4, 8, 11, 1, -1, 6, 14]
for ele1 in l:
for ele2 in l:
if ele1 + ele2 == 10:
print(f'Upon adding {(ele1, ele2)} we get the op 10')
elif ele1-ele2==10:
print(f'Upon subtracting {(ele1, ele2)} we get the op 10')

#-----
'''127. write a decorator to print prefix +91 to the original phone num'''
numbers = [9876543211, 8907654123, 8790456321]
#
def prefix_code(func):
def wrapper(*args, **kwargs):
#
return wrapper
#
@prefix_code
def ph_num(iterable):
for num in iterable:
return num
#
print(ph_num(numbers))
#-----
'''128, wap to get the below output'''
d = {'a':1, 'b':2, 'c':3, 'd':4, 'e':5}
output should be ['b', 'd']
key_list = [ele for ele in d]
print(key_list[1::2]) #['b', 'd']

##Alternate
d = {'a':1, 'b':2, 'c':3, 'd':4, 'e':5}
l = []
for ele in d.keys():
l.append(ele)
print(l[1::2])
#-----
'''129. Can we have multiple init methods in a class'''
class Sample:
#
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
#
sample_obj = Sample(1, 2, 3)
print(sample_obj.a) #1
print(sample_obj.b) #2
print(sample_obj.c) #3
#
sample_obj_2 = Sample(1, 2) #TypeError: __init__() missing 1 required positional
argument: 'c'

#-----
'''130. Why python is object oriented
 ... Because in Python, everything is an object(defined as class)
 ...
#-----
'''131. what are .pyc files '''
#-----
'''132. reverse a list without using any inbuilt funcs and slicing'''

```

```

a = [1, 2, 3, 4, 5]
rev_list = []
for ele in a:
rev_list = [ele] + rev_list
print(rev_list)

##Alternate
a = [10, 20, 30, 40, 50]
rev_list = []
for ele in range(len(a)-1, -1, -1):
rev_list.append(a[ele])
print(rev_list)

##Alternate
a = [10, 20, 30, 40, 50]
rl = [a[ele] for ele in range(len(a)-1, -1, -1)]
print(rl)

#-----
'''134. Difference between while and for loop'''

#-----
'''135. What are magic methods'''

#-----
'''136. Swap two variables without using 3rd variable'''
a = 10
b = 20
b, a = a, b
print(a)
print(b)

#-----
'''137. What is pylint'''

#-----
'''138. What is the output of the following'''
print([1, 2, 3, 4] * 2) #[1, 2, 3, 4, 1, 2, 3, 4]

#-----
'''139. what is the difference between is and == operator '''
#-----
names = ['apple', 'google', 'gmail', 'flipkart', 'yahoo']
for ele in names:
if ele[0] in 'aeiouAEIOU' or ele[-1] in 'aeiouAEIOU':
print(ele)

#-----
a = [1, 2, 3, 4]
b = a[:]
print(b)
print(a == b)
#
print(a is b)

#-----
'''140. What is self in the class'''

#-----
'''141. What is assert statement. what is the difference between assert
and if/else statement'''
assert 3==3
print('hello world')

```



```

#-----
'''142. What is the difference between a module, a package and a library'''
#-----
'''143. wap to get the below output using while loop
1
1 2
1 2 3
1 2 3 4
'''
pat = ''
count = 1
while count <=4:
pat += str(count) + ' '
print(pat)
count += 1

#-----
'''144. wap to get the below output'''
items = ['$123.45', '$434.23', '$567.89']
output should be --> [123.45, 434.23, 567.89]
list_ = []
import re
for ele in items:
res = re.findall(r'[\d\.]+', ele)
for i in res:
list_.append(float(i))
print(list_)

#-----
'''145. function should accept a list and if any number is divisible by 3 then
modify that number to 33, else keep it as it is'''

l = [1, 2, 3, 4, 5, 6, 7, 8, 9]
res = []
for num in l:
if num%3==0:
res += [33]
else:
res += [num]
print(res)

##Alternate Solution
l = [1, 2, 3, 4, 5, 6, 7, 8, 9]
result = [33 if num%3==0 else num for num in l]
print(result)

#-----
'''146. write a fibonacci series'''
def fibo(num):
n1 = 0
n2 = 1
#
for i in range(num):
yield n1
n3 = n2 + n1
n1 = n2
n2 = n3
#
res = fibo(10)
print(res)
for j in res:
print(j)

#-----

```

'''147. wap to print the following pattern

```
1 2 3 *
1 2 * 4
1 * 3 4
* 2 3 4
'''
```

```
pat = ''
index = 4
for ele1 in range(1, 5):
for ele2 in range(1, 5):
if ele2 == index:
print('*', end=' ')
else:
print(ele2, end=' ')
index -= 1
print()
```

#-----