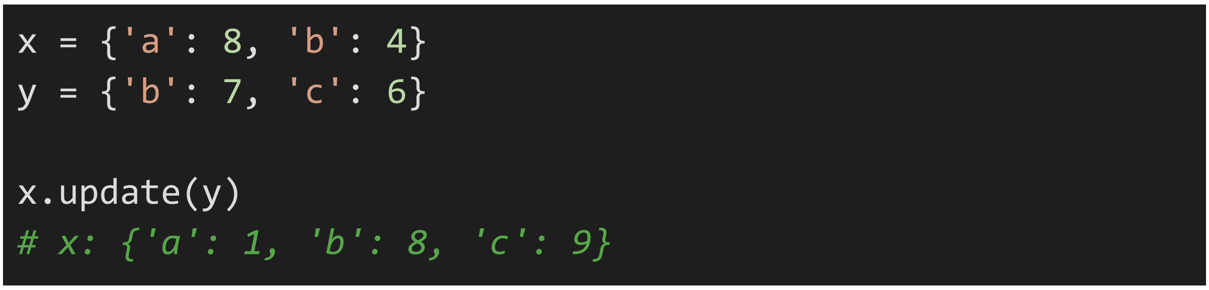
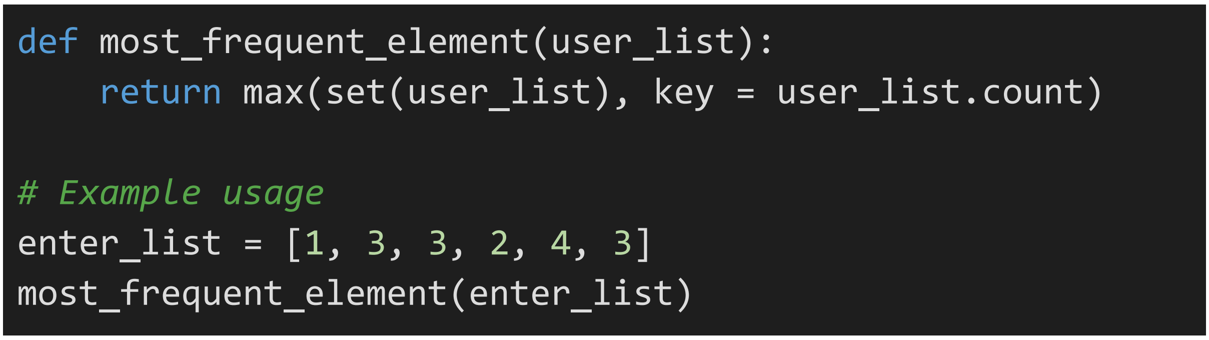
Python_coding_interview_questions

1. How to merge two Dictionaries in Python and write a python code for it.
Answer-



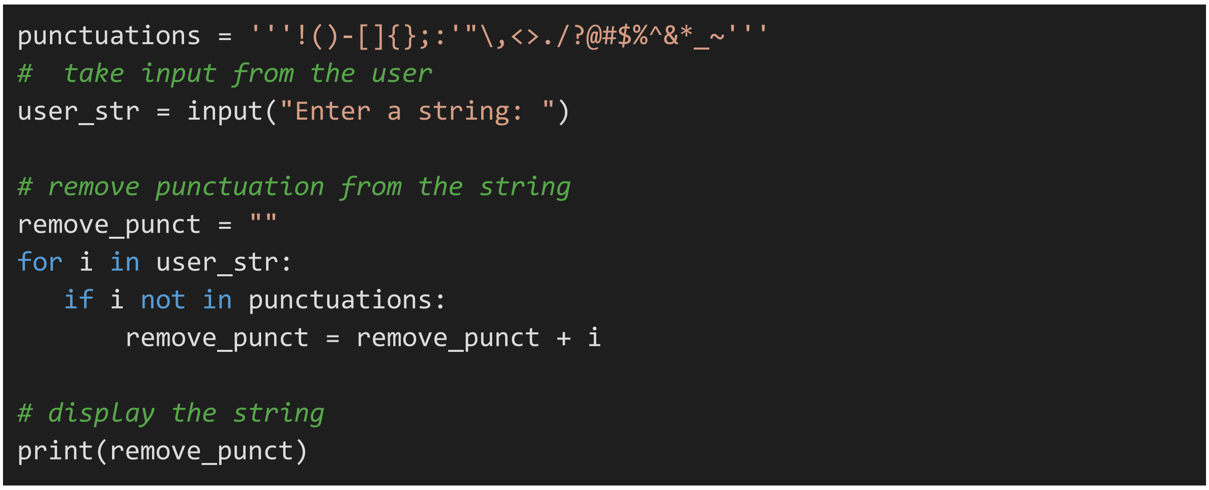
```
x = {'a': 8, 'b': 4}
y = {'b': 7, 'c': 6}

x.update(y)
# x: {'a': 1, 'b': 8, 'c': 9}
```

2. How would you find the most frequent element in a list?
Answer-



```
def most_frequent_element(user_list):
    return max(set(user_list), key = user_list.count)

# Example usage
enter_list = [1, 3, 3, 2, 4, 3]
most_frequent_element(enter_list)
```

3. Write a Python Program to Remove Punctuations From a String.
Answer-



```
punctuations = '''!()-[]{};:'"\,<>./?@#$%^&*_~'''
#  take input from the user
user_str = input("Enter a string: ")

# remove punctuation from the string
remove_punct = ""
for i in user_str:
   if i not in punctuations:
       remove_punct = remove_punct + i

# display the string
print(remove_punct)
```

4. Write a Python Program to Count the Number of Digits in an Integer using a while loop.

```python
Number = 8745
count = 0
while Number != 0:
    Number //= 10
    count += 1
print("Total Number of digits: " + str(count))
```

5. Write a code snippet to generate the square of every list element.
Answer-

```python
user_list = [5,3,6,8,2]
final_list = []
for x in user_list:
    final_list.append(x * x)
print(final_list)
```

6.Write a Python Program for the Sum of squares of first n natural numbers.
Given a positive integer N. The task is to find $1^2 + 2^2 + 3^2 + ... + N^2$.
Answer-

```python
def sum_of_squares(n):
    return sum([i**2 for i in range(1, n+1)])
n = int(input("Enter a number: "))
print("Sum of squares of first", n, "natural numbers:", sum_of_squares(n))
```

7.Write a Python program to interchange the first and last elements in a list.
Input : [1, 2, 3]
Output : [3, 2, 1]
Answer-

```python
def interchange_first_last(lst):
    if len(lst) == 0:
        return []
    else:
        return [lst[-1]] + lst[1:-1] + [lst[0]]

print(interchange_first_last([1,2,3,4,5]))
```

8.Python program to check if a string is a palindrome or not.
Answer-

```
def is_palindrome(string):
    return string == string[::-1]
user_string= input("enter string")
print(is_palindrome(user_string))
```

9. Write a Python Program for the n-th Fibonacci number.
Fn = Fn-1 + Fn-2
Answer-

```
def fibonacci_recursive(n):
    if n <= 0:
        return 0
    elif n == 1:
        return 1
    else:
        return fibonacci_recursive(n-1) + fibonacci_recursive(n-2)
print(fibonacci_recursive(10))
```

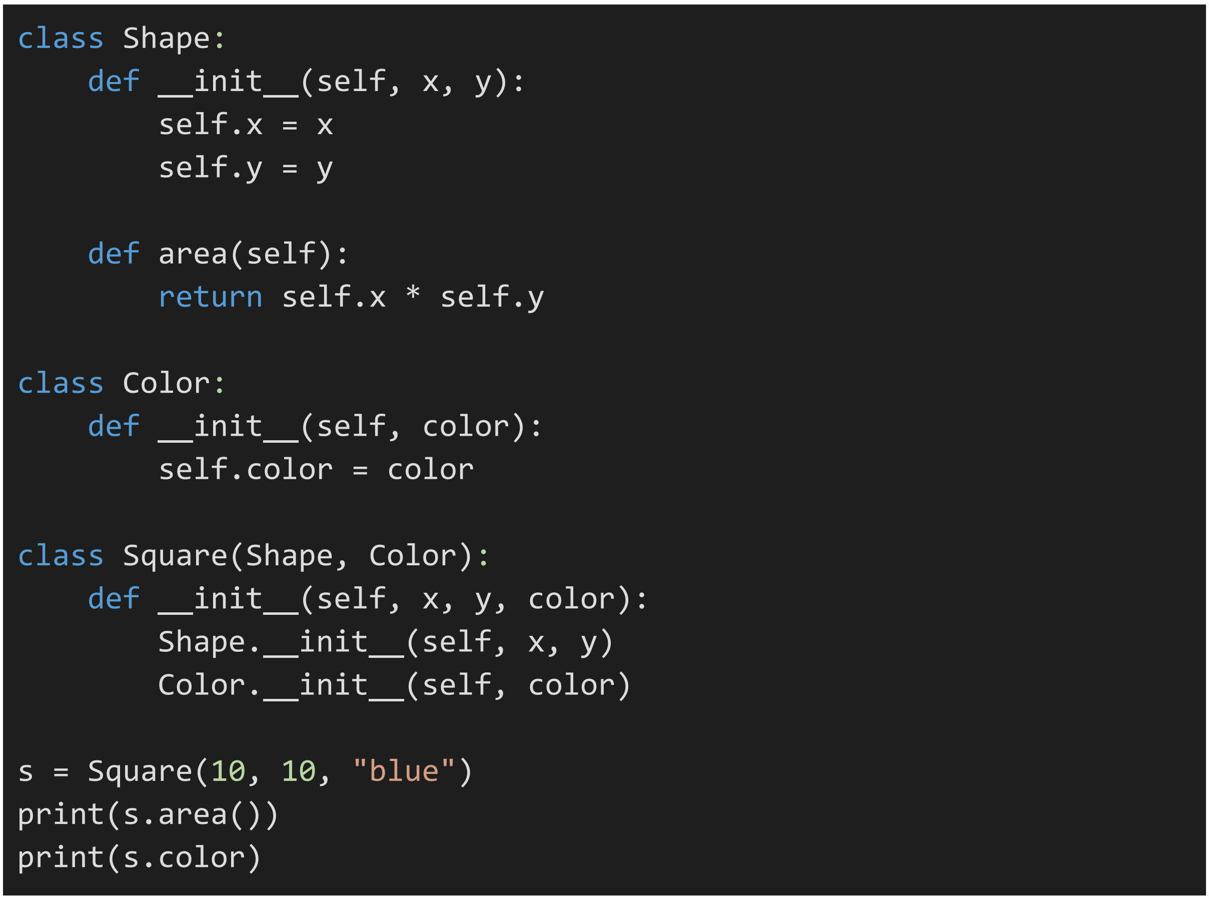
10.Python program to print even-length words in a string.
Answer-

```
def print_even_length_words(s):
    words = s.split()
    even_length_words = [word for word in words if len(word) % 2
== 0]
    return even_length_words

print(print_even_length_words("The quick brown fox jumps over the
lazy dog"))
```

11.How to reverse lists in Python using slicing?
Answer-

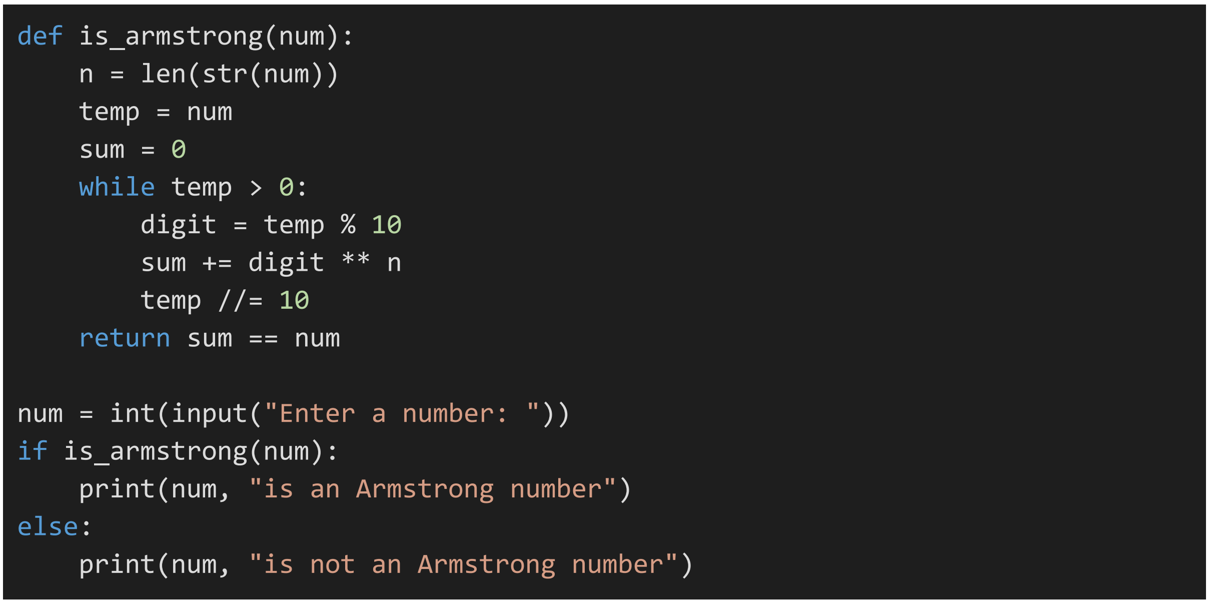
```
numbers = [1, 2, 3, 4, 5]
print(numbers[::-1])
```

12. Write a code to show multiple inheritances in Python.
Answer-



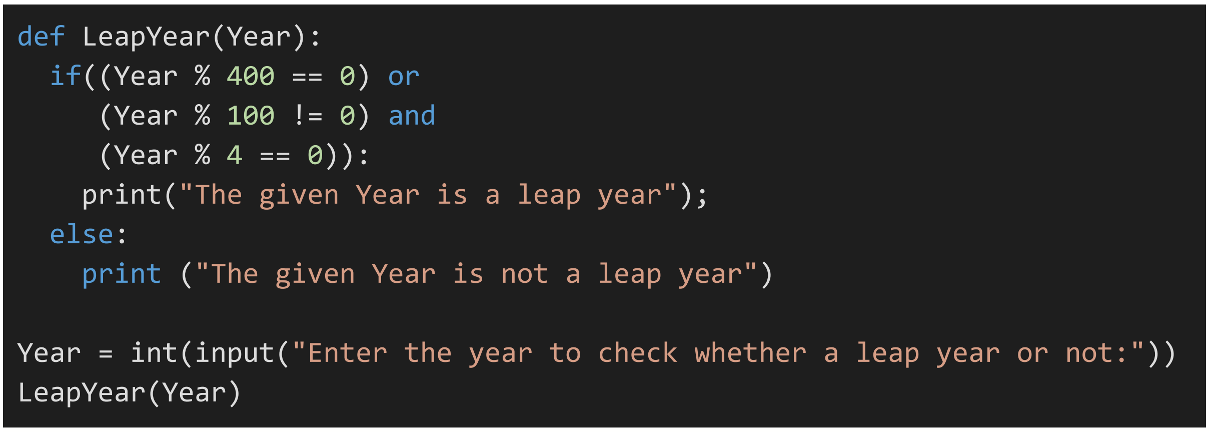
```
class Shape:
    def __init__(self, x, y):
        self.x = x
        self.y = y

    def area(self):
        return self.x * self.y

class Color:
    def __init__(self, color):
        self.color = color

class Square(Shape, Color):
    def __init__(self, x, y, color):
        Shape.__init__(self, x, y)
        Color.__init__(self, color)

s = Square(10, 10, "blue")
print(s.area())
print(s.color)
```

13.Write a Python program to check Armstrong’s number.
Answer-



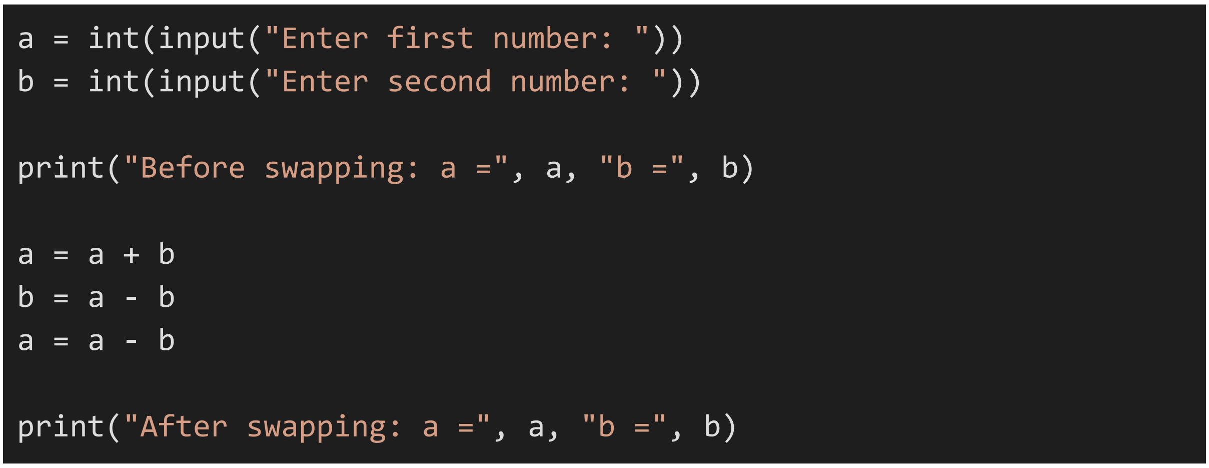
```
def is_armstrong(num):
    n = len(str(num))
    temp = num
    sum = 0
    while temp > 0:
        digit = temp % 10
        sum += digit ** n
        temp //= 10
    return sum == num

num = int(input("Enter a number: "))
if is_armstrong(num):
    print(num, "is an Armstrong number")
else:
    print(num, "is not an Armstrong number")
```

14.Write a Python program to check leap year
Answer-



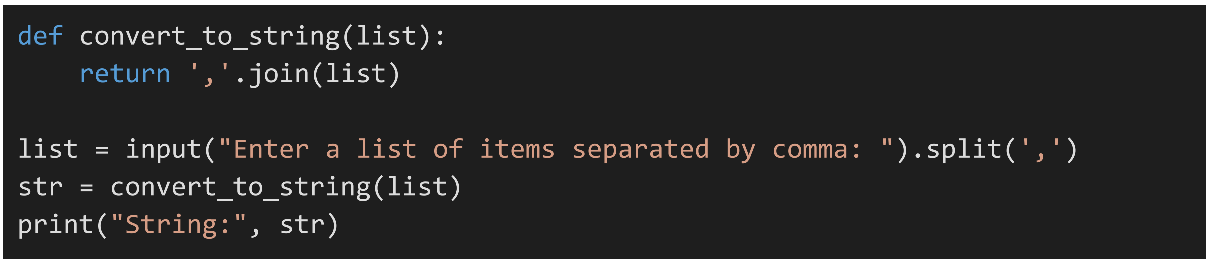
```
def LeapYear(Year):
  if((Year % 400 == 0) or
     (Year % 100 != 0) and
     (Year % 4 == 0)):
    print("The given Year is a leap year");
  else:
    print ("The given Year is not a leap year")

Year = int(input("Enter the year to check whether a leap year or not:"))
LeapYear(Year)
```

15. Write a Python program to swap two variables without using a third variable.
Answer-



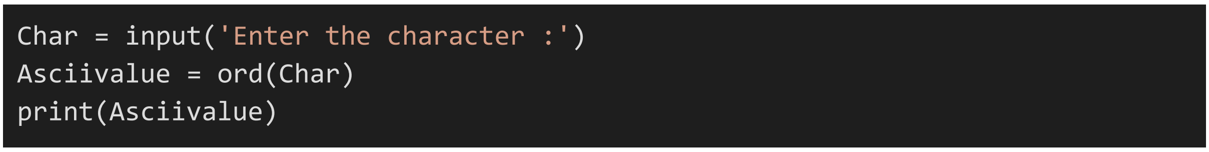
```
a = int(input("Enter first number: "))
b = int(input("Enter second number: "))

print("Before swapping: a =", a, "b =", b)

a = a + b
b = a - b
a = a - b

print("After swapping: a =", a, "b =", b)
```

16.Write a Python Program to Convert Comma Separated List to a String.
Answer-



```
def convert_to_string(list):
    return ','.join(list)

list = input("Enter a list of items separated by comma: ").split(',')
str = convert_to_string(list)
print("String:", str)
```

17.Write a code to find Find the ASCII value of a character.
Answer-



```
Char = input('Enter the character :')
Asciivalue = ord(Char)
print(Asciivalue)
```

18. Write a Python program to get the sum of a non-negative integer.

```
number = int(input("Enter a non-negative integer: "))
sum = 0
while number > 0:
    sum += number % 10
    number //= 10
print("Sum of the digits:", sum)
```

19.Find duplicate numbers in the given array.
arr = [6, 1, 2, 7, 1, 9, 1, 0, 4, 0, 2, 5]
Answer-

```
arr = [6, 1, 2, 7, 1, 9, 1, 0, 4, 0, 2, 5]
frequency = dict()
for ele in arr:
  if ele in frequency:
    frequency[ele] = frequency[ele] + 1
  else:
    frequency[ele] = 1
for i in frequency:
  if frequency[i] > 1:
    print(f"{i} is repeated {frequency[i]} times")
```

20. Write a program to check whether the given two strings are anagrams or not.
Answer-

```
str_1 = "fried"
str_2 = "fired"
if len(str_1) == len(str_2):
  if sorted(str_1) == sorted(str_2):
    print("Given strings are anagrams")
  else:
    print("Given strings are not anagrams")
```

21.Write a Python program that rotates an array by two positions to the right.
Answer-

```
def rotate_array(arr):
    return arr[-2:] + arr[:-2]
array = [1, 2, 3, 4, 5]
print(rotate_array(array))
```

22. Write a Python script to implement the Bubble sort algorithm.

Answer-

```
def bubble_sort(arr):
    n = len(arr)
    for i in range(n):
        for j in range(0, n - i - 1):
            if arr[j] > arr[j + 1]:
                arr[j], arr[j + 1] = arr[j + 1], arr[j]
    return arr
array = [64, 34, 25, 12, 22, 11, 90]
print("Sorted array is:", bubble_sort(array))
```

23. Merge two dictionaries in a single expression.

Answer-

```
data_science_students = {1: 'vinu', 2: "sudhanshu", 3:"prachi"}
web_development_students  = {2: 'sudhanshu', 4: "rohit"}

allstudents = {**data_science_students, **web_development_students}
print(allstudents)
```

24. Write a program that will find all such numbers which are divisible by 7 but are not a multiple of 5, between 2000 and 3200 (both included).

The numbers obtained should be printed in a comma-separated sequence on a single line.

Answer-

```
result = []
for i in range(2000, 3201):
    if i % 7 == 0 and i % 5 != 0:
        result.append(str(i))
print(','.join(result))
```

25. Print the following number pattern.

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

Answer-

```
rows = 5
x = 0
for i in range(rows, 0, -1):
    x += 1
    for j in range(1, i + 1):
        print(x, end=' ')
    print('\r')
```

26. Find the maximum values in a given heterogeneous list using lambda.
Answer-

```
def max_value(list_val):
     maximum_value = max(user_list, key = lambda i: (isinstance(i, int), i))
     return(maximum_value)

user_list = [ 7, 2,'Python', 1, 8, 'version']
print("Original list is:")
print(user_list)
print("\nMaximum values in the list using lambda:")
print(max_value(user_list))
```

27.Write a Python program to find a list with maximum and minimum length using lambda.
Answer-

```
def max_length_list(input_list):
    max_length = max(len(x) for x in input_list )
    max_list = max(input_list, key = lambda i: len(i))
    return(max_length, max_list)

def min_length_list(input_list):
    min_length = min(len(x) for x in input_list )
    min_list = min(input_list, key = lambda i: len(i))
    return(min_length, min_list)

list1 = [[0], [13, 15, 17], [1, 3], [5, 7], [9, 11]]
print("Original list:")
print(list1)
print("\nList with maximum length of lists:")
print(max_length_list(list1))
print("\nList with minimum length of lists:")
print(min_length_list(list1))
```

28. Write a Python program to add two given lists using map and lambda.
Original list:
[1, 2, 3]
[4, 5, 6]
Result: after adding two list
[5, 7, 9]
Answer-

```
list1 = [1, 2, 3]
list2 = [4, 5, 6]
print("Original lists:")
print(list1)
print(list2)
result = map(lambda x, y: x + y, list1, list2)
print("After adding two list")
print(list(result))
```

29.Write a Python program to add three given lists using Python map and lambda.
list1 = [1, 2, 3]
list2 = [4, 5, 6]
list3 = [7, 8, 9]
Answer-

```
list1 = [1, 2, 3]
list2 = [4, 5, 6]
list3 = [7, 8, 9]
result = map(lambda x, y, z: x + y + z, list1, list2, list3)
print("lists after adding above three lists:",list(result))
```

30.Write a Python program to calculate the value of 'x' to the power of 'y''.
Answer-

```
def power(x,y):
  if y==0:
    return 1
  elif x==0:
    return 0
  elif y==1:
    return x
  else:
    return x*power(x,y-1)
print(power(5,4))
```

31.Write a Python program to find the greatest common divisor (GCD) of two integers.

Answer-

```
def Recurgcd(a, b):
  low = min(a, b)
  high = max(a, b)
  if low == 0:
    return high
  elif low == 1:
    return 1
  else:
    return Recurgcd(low, high%low)
print(Recurgcd(40,20))
```

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

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

Answer-

```
n=5;
for i in range(n):
    for j in range(i):
        print ('* ', end="")
    print('')
for i in range(n,0,-1):
    for j in range(i):
        print('* ', end="")
    print('')
```

33.Write a Python program to print alphabet pattern 'A'.
Answer-

```
result_str="";
for row in range(0,7):
    for column in range(0,7):
        if (((column == 1 or column == 5) and row != 0) or ((row == 0 or row == 3) and (column > 1 and column < 5))):
            result_str=result_str+"*"
        else:
            result_str=result_str+" "
    result_str=result_str+"\n"
print(result_str)
```

34. Write a Python program to construct the following pattern, using a nested loop number.
Expected Output:

```
1
22
333
4444
55555
666666
7777777
88888888
999999999
```

Answer-

```
for i in range(10):
    print(str(i) * i)
```

35. Write a Python program to print alphabet pattern 'T'.
Expected Output:

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

Answer-

```
result_str=""
for row in range(0,7):
    for column in range(0,7):
        if (column == 3 or (row == 0 and column > 0 and column <6)):
            result_str=result_str+"*"
        else:
            result_str=result_str+" "
    result_str=result_str+"\n"
print(result_str)
```

36. Write a Python program to get next day of a given date.
Answer-

```
year = int(input("Input a year: "))
if (year % 400 == 0):
    leap_year = True
elif (year % 100 == 0):
    leap_year = False
elif (year % 4 == 0):
    leap_year = True
else:
    leap_year = False
month = int(input("Input a month [1-12]: "))
if month in (1, 3, 5, 7, 8, 10, 12):
    month_length = 31
elif month == 2:
    if leap_year:
        month_length = 29
    else:
        month_length = 28
else:
    month_length = 30
day = int(input("Input a day [1-31]: "))
if day < month_length:
    day += 1
```

```python
else:
    day = 1
    if month == 12:
        month = 1
        year += 1
    else:
        month += 1
print("The next date is [yyyy-mm-dd] %d-%d-%d." % (year, month, day))
```

37. Write a Python program to unpack a tuple into several variables.
Answer-

```python
tuplex = 4, 8, 3
print(tuplex)
n1, n2, n3 = tuplex
#unpack a tuple in variables
print(n1 + n2 + n3)
#the number of variables must be equal to the number of items of the tuple
n1, n2, n3= tuplex
```

38. Write a Python program to add an item to a tuple.
Answer-

```python
tuplex = (4, 6, 2, 8, 3, 1)
print(tuplex)
#tuples are immutable, so you can not add new elements
#using merge of tuples with the + operator you can add an element and it will create a new tuple
tuplex = tuplex + (9,)
print(tuplex)
#adding items in a specific index
tuplex = tuplex[:5] + (15, 20, 25) + tuplex[:5]
print(tuplex)
#converting the tuple to list
listx = list(tuplex)
#use different ways to add items in list
listx.append(30)
tuplex = tuple(listx)
print(tuplex)
```

39.Write a Python program to remove an item from a tuple.
Answer-

```
tuplex = "w", 3, "r", "s", "o", "u", "r", "c", "e"
print(tuplex)
#tuples are immutable, so you can not remove elements
#using merge of tuples with the + operator you can remove an item and it
will create a new tuple
tuplex = tuplex[:2] + tuplex[3:]
print(tuplex)
#converting the tuple to list
listx = list(tuplex)
#use different ways to remove an item of the list
listx.remove("c")
#converting the tuple to list
tuplex = tuple(listx)
print(tuplex)
```

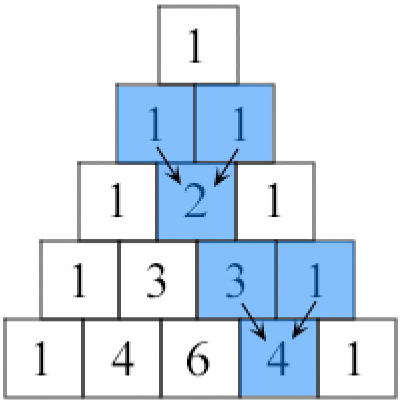
40.Write a Python program to get the frequency of the tuples in a given list.
Answer-

```
from collections import Counter
nums = [(['1', '4'], ['4', '1'], ['3', '4'], ['2', '7'], ['6', '8'],
['5','8'], ['6','8'], ['5','7'], ['2','7'])]
print("Original list of tuples:")
print(nums)
result = Counter(tuple(sorted(i)) for i in nums[0])
print("\nTuples","    ","frequency")
for key,val in result.items():
    print(key," ", val)
```

41. Write a Python program to sum recursion lists.
Test Data: [1, 2, [3,4], [5,6]]
Expected Result: 21
Answer-

```
def recursive_list_sum(data_list):
  total = 0
  for element in data_list:
    if type(element) == type([]):
      total = total + recursive_list_sum(element)
    else:
      total = total + element

  return total
print( recursive_list_sum([1, 2, [3,4],[5,6]]))
```

42. Write a Python function that prints out the first n rows of Pascal's triangle.



Answer-

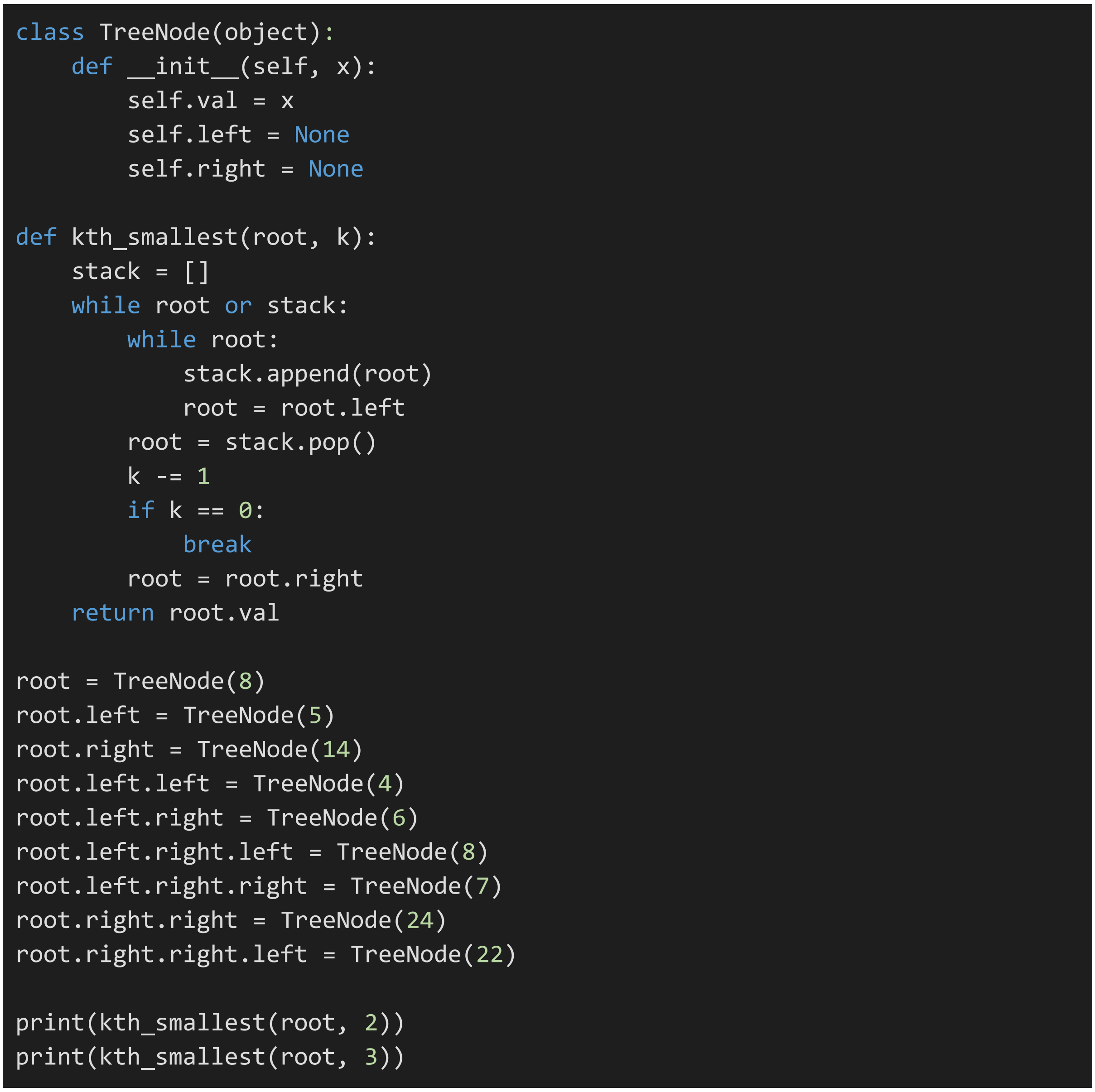
```
def pascal_triangle(n):
   trow = [1]
   y = [0]
   for x in range(max(n,0)):
      print(trow)
      trow=[l+r for l,r in zip(trow+y, y+trow)]
   return n>=1
pascal_triangle(6)
```

43. Write a Python program to execute a string containing Python code.

```
mycode = 'print("hello world")'
code = """
def mutiply(x,y):
    return x*y

print('Multiply of 2 and 3 is: ',mutiply(2,3))
"""
exec(mycode)
exec(code)
```

44. Write a Python program to find the kth smallest element in a given binary search tree.

Answer-



```
class TreeNode(object):
    def __init__(self, x):
        self.val = x
        self.left = None
        self.right = None

def kth_smallest(root, k):
    stack = []
    while root or stack:
        while root:
            stack.append(root)
            root = root.left
        root = stack.pop()
        k -= 1
        if k == 0:
            break
        root = root.right
    return root.val

root = TreeNode(8)
root.left = TreeNode(5)
root.right = TreeNode(14)
root.left.left = TreeNode(4)
root.left.right = TreeNode(6)
root.left.right.left = TreeNode(8)
root.left.right.right = TreeNode(7)
root.right.right = TreeNode(24)
root.right.right.left = TreeNode(22)

print(kth_smallest(root, 2))
print(kth_smallest(root, 3))
```

45. Write a Python program to multiply two integers without using the * operator.
Answer-

```
def multiply(x, y):
    if y < 0:
        return -multiply(x, -y)
    elif y == 0:
        return 0
    elif y == 1:
        return x
    else:
        return x + multiply(x, y - 1)
print(multiply(3, 5))
```

46.Write a Python program to flip a coin 1000 times and count heads and tails.
Expected Output :
Heads: 5073
Tails: 4927
Answer-

```
import random
import itertools

results = {
    'heads': 0,
    'tails': 0,
}
sides = list(results.keys())

for i in range(10000):
    results[random.choice(sides)] += 1
print('Heads:', results['heads'])
print('Tails:', results['tails'])
```

47. Write a program which can filter even numbers in a list by using filter function.
The list is: [1,2,3,4,5,6,7,8,9,10].
Answer-

```
li = [1,2,3,4,5,6,7,8,9,10]
evenNumbers = filter(lambda x: x%2==0, li)
print(evenNumbers)
for i in evenNumbers:
  print(i)
```

48.By using list comprehension, please write a program generate a 3*5*8 3D array whose each element is 0.
Answer-

```
array = [[ [0 for col in range(8)] for col in range(5)] for row in range(3)]
print(array)
```

49. Please write a binary search function which searches an item in a sorted list. The function should return the index of element to be searched in the list.
Answer-

```
import math
def bin_search(li, element):
    bottom = 0
    top = len(li)-1
    index = -1
    while top>=bottom and index==-1:
        mid = int(math.floor((top+bottom)/2.0))
        if li[mid]==element:
            index = mid
        elif li[mid]>element:
            top = mid-1
        else:
            bottom = mid+1

    return index

li=[2,5,7,9,11,17,222]
print bin_search(li,11)
print bin_search(li,12)
```

50. Write a Python program to find the next smallest palindrome of a specified number.
Answer-

```
import sys
def Next_smallest_Palindrome(num):
    numstr = str(num)
    for i in range(num+1,sys.maxsize):
        if str(i) == str(i)[::-1]:
            return i
print(Next_smallest_Palindrome(99))
print(Next_smallest_Palindrome(1221))
```