# PYTHON QUESTIONS

1.Create a string made of the first, middle and last character

2. Create a string made of the middle three characters

3. Append new string in the middle of a given string

4. Create a new string made of the first, middle, and last characters of each input string

5. Arrange string characters such that lowercase letters should come first

6. Count all letters, digits, and special symbols from a given string

8. String characters balance Test

9. Find all occurrences of a substring in a given string by ignoring the case

10. Calculate the sum and average of the digits present in a string

11. How to find a substring in a given string?

12. How to find a substring in a given string using string slicing

13. What will be the code block, if we can’t find a substring within a string.

14. Remove empty strings from the list of strings

15. Add new item to list after a specified item

16. Extend nested list by adding the sub list

17. Replace list’s item with new value if found

18. Remove all occurrences of a specific item from a list.

19. Print 2nd highest number.

20. Print the lowest number

21. What Is the Difference Between Del and Remove() on Lists?

22. Differentiate Between append() and extend()

23. var=”Red, Blue, Green, Orange” lst=var.split(‘,’,0) | What will be the output of the program

24. Can we perform multiplication of 2 tuples. Write code block

25. What Does the ‘is’ Operator Do?

26. How can we convert a list into dictionary in python? Write code block.

27. Please mention the output

str1 = “goose”

a = len(str1)

print (a^2)

28. How can we convert a set into dictionary? Write code block.

29. What is the Complexity of Algorithm?

30. Write pseudocode for: fn = f1 + (n-1)d

31. Find the length of a tuple. 3

32. Concatenate two tuples.

33. Access an element in a tuple.

34. Count the occurrences of an element in a tuple.

35. Check if an element exists in a tuple.

36. Convert a tuple to a string.

37. Find the index of an element in a tuple.

38. Convert a list to a tuple.

39. Iterate through a tuple using a loop.

40. Find the maximum and minimum elements in a tuple.

41. Convert a tuple of strings to a single string.

42. Remove an element from a tuple.

43. Find the common elements between two tuples.

44. Sort a tuple of integers.

45. Find the sum of all elements in a tuple.

46. Merge two tuples and remove duplicates.

47. Find the first and last elements of a tuple

48. Convert a tuple of integers to a tuple of strings.

49. Count the number of even and odd numbers in a tuple.

50. Find the product of all elements in a tuple

51. Write a program to demonstrate a simple calculator using function

52. Write a program to demonstrate factorial using recursion function.

53. Write a program to demonstrate the factorial using recursion.

54. Write a program to demonstrate multiplication of 2 lists elements using for loop

55. Write a program to demonstrate square of every alternate list element and append it to another list using for loop.

56. Write a program to remove duplicate elements from a list.

57. Write a program to find the number of occurrences of an element in a numeric only list and find the sum of that element and store it in another list.

58. Reverse the tuple

59. Access value 20 from the tuple

60. Create a tuple with single item 50

61. Unpack the tuple into 4 variables

62. Swap two tuples in Python

63. Copy specific elements from one tuple to a new tuple

64. Modify the tuple

65. Counts the number of occurrences of item 50 from a tuple

66. Check if all items in the tuple are the same

67. Python program to remove empty tuples from a List

Input: tuples = [(), (‘r’,’15’,’8′), (), (‘l’, ‘’), (‘k’, ‘a’, ’45’), (”,”),()]

Output: [(‘r’, ’15’, ‘8’), (‘l’, ‘s’), (‘k’, ‘a’, ’45’), (”, ”)]

68. Write the code for the following

The original list: [4, 5, 6]

The list after alternates repeating elements: [4, 4, 4, 6, 6, 6]

69. Write the code for the following

Input: test\_list = [“Gfg”, 3], key\_list = [“name”, “id”]

Output: [{‘name’: ‘Gfg’, ‘id’: 3}

70. Write the code for the following

Input: set ([12, 10, 13, 15, 8, 9])

Output:

{9, 10, 12, 13, 15}

{10, 12, 13, 15}

{12, 13, 15}

{13, 15}

{15}

set()

71. Write code to convert any base to decimal by using int().

Input: '1011'

base = 2

Output: 11

72. Create a list and perform the following methods

1) insert () 2) remove () 3) append () 4) len() 5) pop() 6) clear()

73. Write a python program to find the length of list?

74. Create a tuple and perform the following methods

1) Add items 2) len() 3) check for item in tuple 4)Access items

75. Write a python program using the following methods: 1) count 2) index

76. Write a python program using “+” and “\*” operations which resulting a new tuple?

77. Create a dictionary and apply the following methods

1) Print the dictionary items 2) access items 3) use get() 4)change values 5) use len()

78. Write a python code to convert list of tuples into dictionaries?

79. Write python program to store data in list, tuple, set, dictionary and then try to print them.

80. Write a Python program to perform the basic four operators (+, -, \*, /)

81. Write a python Program to read a number and display corresponding day using if\_elif\_else?

## ALGORITHIM WRITING Write the output of the program

1. class Clock(object):

def \_init\_(self, time):

self.time = time

def print\_time(self):

self.time = "6:30"

print(self.time)

clock = Clock("5:30")

clock.print\_time()

1. class Clock(object):

def \_init\_(self, time):

self.time = time

def print\_time(self, time):

print(self.time)

clock = Clock("5:30")

clock.print\_time("10:30")

3.

class Spell(object):

def \_init\_(self, incantation, name):

self.name = name

self.incantation = incantation

def \_str\_(self):

return self.name + " " + self.incantation + "\n" + self.get\_description()

def get\_description(self):

return "No description"

def execute(self):

print(self.incantation)

class Accio(Spell):

def \_init\_(self):

Spell.\_init\_(self, "Accio", "Summoning Charm")

class Confundo(Spell):

def \_init\_(self):

Spell.\_init\_(self, "Confundo", "Confundus Charm")

def get\_description(self):

return "Causes the victim to become confused and befuddled."

def study\_spell(spell):

print(spell)

spell = Accio()

spell.execute()

study\_spell(spell)

study\_spell(Confundo())