## Assignment Part-1

Q1. Why do we call Python as a general purpose and high-level programming language?

A1. Python is called a general purpose language because it has application in varied domains. Unlike a domain-specific language (DSL), it is not specified for a particular domain and is a beginner-friendly language as well. It is considered high -level programming language as it has been developed for easy understanding of humans and not written in machine-readable format, that is, low-level language.

Q2. Why is Python called a dynamically typed language?

A2. Python is called a dynamically type language because the type of the variable is assigned only during the runtime. There is no nessecity to declare the type of the variable in the written code.

Q3. List some pros and cons of Python programming language?

A3. Some pros of Python programming language are:

a. It is a beginner-friendly language as it is easy to read and use by humans

b. Python developers have made it open-source for everyone to use it for free

c. A large community to support anyone who needs insights

d. It has a huge collection of libraries for different application

e. It is portable and scalable programming language

Some cons of Python programming language are:

a. It is slower than compiled languages and thus, one major drawback is the speed limitation

b. It does not support much of mobile environment

c. It has a huge memory consumption

d. It shows runtime error for its dynamically typed feature

Q4. In what all domains can we use Python?

A4. Python has a wide variety of domains applicability, such as, data engineering, data science, web development, machine learning/artificial intelligence, mobile app development, etc.

Q5. What are variable and how can we declare them?

A5. A variable is used to store values or information in the memory. It is a reference that can retrieve stored values from the memory location as and when required. It can be declared by writing the type before the name of the variable and assigning it a value. In Python, mentioning a type is not needed, it gets declare just through assignment of the value.

Q6. How can we take an input from the user in Python?

A6. Input can be taken by using the function 'input()' and putting a string into the prompt of the function. For example: x = input("Enter the value of x:").

Q7. What is the default datatype of the value that has been taken as an input using input() function?

A7. The input() function as string as its default datatype of the value.

Q8. What is type casting?

A8. Type casting is the method of converting one data type to another datatype. For example, x = 2 is int type and then changing it to y = float(x) with y as float.

Q9. Can we take more than one input from the user using single input() function? If yes, how? If no, why?

A9. No, we can take only one input from the user using single input() function as only one value can be assigned to the variable for which the input is being taken. To take more than one value, another function split() needs to be used along with the input() function.

Q10. What are keywords?

A10. Python keywords are reserved words that have specific purposes to define the syntax of the language and cannot be used for any other purpose.

Q11. Can we use keywords as a variable? Support your answer with reason.

A11. No, keywords cannot be used as a variable because they have special meanings. The keywords, such as 'and', 'or', 'if', etc. are logical operators that consturct the structure needed to implement the python language.

Q12. What is indentation? What's the use of indentaion in Python?

A12. Indentation is the spaces put at the begining of a codee line. It is very important in Python as it defines a block of code, that is, a group of statements that performs a particular task together.

Q13. How can we throw some output in Python?

A13. To get output in Python, the most commonly used function is print().

Q14. What are operators in Python?

A14. Operators in Python are used to perform mathematical operations (+, -, \*, etc.), assign values to variable (=, +=, /=, etc.), compare variables and return a boolean result (True, False).

Q15. What is difference between / and // operators?

A15. Operator / is used for float division whereas // is used for interger divsion.

Q16. Write a code that gives following as an output.

```

iNeuroniNeuroniNeuroniNeuron

```

A16.

str\_data = 'iNeuron'

str\_result = str\_data \* 4

print("'''\n", str\_result, "\n'''")

Q17. Write a code to take a number as an input from the user and check if the number is odd or even.

A17.

num = int(input("Enter a number: "))

if num % 2 == 0:

print("The number", num, "is even.")

else:

print("The number", num, "is odd.")

Q18. What are boolean operator?

A18. Boolean operators are the keywords 'and', 'or', 'not' that are used in various combination to perform some logical operations. True and False are also considered boolean operators.

Q19. What will the output of the following?

```

1 or 0

0 and 0

True and False and True

1 or 0 or 0

```

A19: Output:

1

0

False

1

Q20. What are conditional statements in Python?

A20. Conditional statements in Python are used for decision-making operations. The conditional operators run a particular block of code only when a certain condition is satisfied.

Q21. What is use of 'if', 'elif' and 'else' keywords?

A21. 'if', 'elif' and 'else' keywords are used to check multiple expressions. The first condition begins with 'if', the next set of conditions begin with 'elif' and the last condition or whatever is left is put to 'else' keyword.

Q22. Write a code to take the age of person as an input and if age >= 18 display "I can vote". If age is < 18 display "I can't vote".

A22.

age = int(input("Enter the age of a person: "))

if age >= 18:

print("I can vote")

else:

print("I can't vote")

Q23. Write a code that displays the sum of all the even numbers from the given list.

```

numbers = [12, 75, 150, 180, 145, 525, 50]

```

A23:

numbers = [12, 75, 150, 180, 145, 525, 50]

sum = 0

for num in numbers:

if num % 2 == 0:

sum += num

print("The sum of all the even numbers from the given list is", sum)

Q24. Write a code to take 3 numbers as an input from the user and display the greatest no as output.

A24:

numbers = input("Enter three numbers: ")

num = [int(n) for n in numbers.split()]

m = max(num)

print("The greatest number is:", m)

Q25. Write a program to display only those numbers from a list that satisfy the following conditions

- The number must be divisible by five

- If the number is greater than 150, then skip it and move to the next number

- If the number is greater than 500, then stop the loop

```

numbers = [12, 75, 150, 180, 145, 525, 50]

```

A25.

numbers = [12, 75, 150, 180, 145, 525, 50]

num = []

for n in numbers:

if n > 150:

if n > 500:

break

continue

if n % 5 == 0:

num.append(n)

print("The list of numbers that satisfy the conditions:", num)

Q26. What is a string? How can we declare string in Python?

A26. String is a sequence of characters, alphabets or words. To declare string in Python, we can put the sequence of characters inside either single quotes, double quotes or triple quotes and then assign it to a variable.

Q27. How can we access the string using its index?

A27. We can access the characters in a string by referring to its index number inside square brackets.

Q28. Write a code to get the desired output of the following

string = "Big Data iNeuron"

desired\_output = "iNeuron"

A28. string = "Big Data iNeuron"

des\_output = string[9:]

print('desire output = "', des\_output,'"')

Q29. Write a code to get the desired output of the following

string = "Big Data iNeuron"

desired\_output = "norueNi"

A29. string = "Big Data iNeuron"

des\_output = string[:8:-1]

print('desire output = "', des\_output,'"')

Q30. Reverse the string given in the above question.

A30. string = "Big Data iNeuron"

des\_output = string[::-1]

print('desire output = "', des\_output,'"')

Q31. How can you delete entire string at once?

A31. To delete an entire string at once, one can assign an empty string or use the del statement.

Q32. What is escape sequence?

A32. Escape sequence is the insertion of characters that are illegal in a string and never get printed as part of the output.

Q33. How can you print the below string?

'iNeuron's Big Data Course'

A33. string = "iNeuron's Big Data Course"

print(string)

Q34. What is a list in Python?

A34. Lists is one of the built-in data types that is used to store multiple items in a single variable.

Q35. How can you create a list in Python?

A35. To create a list in Python, we can assign a collection of data within square brackets to a variable or just assign empty square brackets to a variable.

Q36. How can we access the elements in a list?

A36. We can use the square brackets for slicing along the index or indices to access the elements in a list.

Q37. Write a code to access the word "iNeuron" from the given list.

lst = [1,2,3,"Hi",[45,54, "iNeuron"], "Big Data"]

A37. lst = [1,2,3,"Hi",[45,54, "iNeuron"], "Big Data"]

w = lst[4][2]

print(w)

Q38. Take a list as an input from the user and find the length of the list.

A38. n = input("Enter elements of the list sewparated by space: ")

lst = n.split()

l = len(lst)

print("The length of the list is ", l)

Q39. Add the word "Big" in the 3rd index of the given list.

lst = ["Welcome", "to", "Data", "course"]

A39. lst = ["Welcome", "to", "Data", "course"]

lst.insert(2, "Big")

print("The list is ", lst)

Q40. What is a tuple? How is it different from list?

A40. Tuple is one of the built-in data types in Python that is used to store multiple items in a single variable. It is immutable unlike the list, that is, the elements of a tuple cannot be changed.

Q41. How can you create a tuple in Python?

A41. A tuple is created by placing all the items within parentheses (optional) that are separated by commas.

Q42. Create a tuple and try to add your name in the tuple. Are you able to do it? Support your answer with reason.

A42. No, another element (my name) cannot be added to a tuple. Tuples are immutable in nature and there is no append() or extend() functions for tuples.

Q43. Can two tuple be appended. If yes, write a code for it. If not, why?

A43. Yes, two tuples can be appended using concatenation.

Code: t1 = (3, 5, 2, 30)

t2 = t1 + (8, 18, 2)

print(t2)

print(t1)

Q44. Take a tuple as an input and print the count of elements in it.

A44. t1 = (3, 5, 2, 30)

l = len(t1)

print(l)

Q45. What are sets in Python?

A45. Sets are one of the 4 built-in data types that are used to store multiple items in a single variable. It is a collection which is unordered.

Q46. How can you create a set?

A46. Sets are created by putting all the elements inside curly braces {}, separated by comma. Set can have different data types in it and is mutable in nature.

Q47. Create a set and add "iNeuron" in your set.

A47. s1 = {4, "flower", 37}

s1.add("iNeuron")

print(s1)

Q48. Try to add multiple values using add() function.

A48. s1 = {4, "flower", 37}

s1.update(["iNeuron", "Big Data", 21])

print(s1)

Q49. How is update() different from add()?

A49. The update() is used to append multiple values to the set whereas add() appends a single item to the set.

Q50. What is clear() in sets?

A50. The clear() method that is used to remove all elements from the set.

Q51. What is frozen set?

A51. The frozen set is an immutable version of sets in python. It is a built-in function in Python.

Q52. How is frozen set different from set?

A52. The elements of a set can be modified at any time whereas the elements of the frozen set remain the same after creation.

Q53. What is union() in sets? Explain via code.

A53. The union() function returns a set that contains all elements from both the original sets and no item is repeated.

Code: A = {15, 43, 23, 71}

B = {1, 23, 115}

print(A.union(B))

#output = {1, 115, 71, 23, 43, 15}

Q54. What is intersection() in sets? Explain via code.

A54. The interaction() function returns a set that contains the common items from both original sets.

Code: A = {15, 43, 23, 71}

B = {1, 23, 115}

print(A.intersection(B))

#output = {23}

Q55. What is dictionary ibn Python?

A55. Dictionaries are a collection of key values, used to store data values like a map. They hold key:value pair.

Q56. How is dictionary different from all other data structures.

A56. The dictionary data structure in Python is an unordered collection of items with keys mapping to its values. While other data structures use only one value as an element. Thus, dictionary is more compound in form than the other data structures.

Q57. How can we delare a dictionary in Python?

A58. A dictionary can be created by placing a sequence of elements within curly brace {}, separated by comma. Dictionary holds pair of items, key:value.

Q58. What will the output of the following?

var = {}

print(type(var))

A58. <class ‘dict’>.

Q59. How can we add an element in a dictionary?

A59. We can add an element to a dictionary by inserting a new index key into the dictionary, then assigning it a particular value.

Q60. Create a dictionary and access all the values in that dictionary.

A60. Fruits = {1: 'Apple', 2: 'Orange', 3: 'Peach'}

val\_Fruits = Fruits.values()

print(val\_Fruits)

Q61. Create a nested dictionary and access all the element in the inner dictionary.

A61. Dict = { 'Fruits' : {1: 'Apple', 2: 'Orange', 3: 'Peach'},

'Flowers': {'Color': 'Pink', 1: ['Hibiscus', 5]} }

ele\_Dict = Dict.values()

print(ele\_Dict)

Q62. What is the use of get() function?

A62. The get() function returns the value of the item with specific key.

Q63. What is the use of items() function?

A63. The items() function is used to return key-value pairs of the dictionary.

Q64. What is the use of pop() function?

A64. The pop() function is used to remove an item at the specified index from the list.

Q65. What is the use of popitems() function?

A65. The popitem() function removes the element that was last inserted into the dictionary.

Q66. What is the use of keys() function?

A66. The keys() function is used to return a list of all the keys in the dictionary in order of insertion using Python.

Q67. What is the use of values() function?

A67. The values() function is used to return a list of all the values in the dictionary.

Q68. What are loops in Python?

A68. A loop is used to repeat/iterate something over and over again until a particular condition is satisfied.

Q69. How many type of loop are there in Python?

A69. There are mainly 2 types of loop in Python. The for and while loop. When loop is used inside another loop, it becomes nested loop.

Q70. What is the difference between for and while loops?

A70. For loops are used to iterate over sequence like list, tuple, set, etc. whereas while loops are used to execute a set of statements if a condition is true.

Q71. What is the use of continue statement?

A71. The continue statement is used to return the control to the beginning of the loop. The continue statement ignores the rest of the statements in the current iteration of the loop and moves the control back to the top of the loop.

Q72. What is the use of break statement?

A72. The break statement terminates the current loop and resumes execution at the next statement.

Q73. What is the use of pass statement?

A73. The pass statement is used as a placeholder for future code. When the pass statement is executed, nothing happens, but one avoids getting an error when empty code is not allowed.

Q74. What is the use of range() function?

A74. The range() function is used return a sequence of numbers, starting from 0 by default, and is incremented by 1 (by default), and stop before a specified number.

Q75. How can you loop over a dictionary?

A75. To loop through a dictionary, one can use a for loop. When looping through a dictionary, the return value are the keys of the dictionary, but there are methods to return the values as well.

**Coding problems**

Q76. Write a Python program to find the factorial of a given number.

A76. def factorial(n):

num = 1

while n >= 1:

num = num \* n

n = n - 1

return num

x = int(input("Enter a number for factorial:\n"))

print(factorial(x))

Q77. Write a Python program to calculate the simple interest. Formula to calculate simple interest is SI = (P*R*T)/100

A77. def interest(p, r, t):

si = 1

si = (p \* r \* t)/100

return si

x, y, z = input("Enter the values for P, R, T:\n").split()

p = int(x)

t = int(z)

r = int(y)

print("The simple interst is: ", interest(p, r, t))

Q78. Write a Python program to calculate the compound interest. Formula of compound interest is A = P(1+ R/100)^t.

A78. def interest(p, r, t):

a = 1

a = p \* ((1 + (r/100)) \*\* t)

return a

x, y, z = input("Enter the values for P, R, T:\n").split()

p = int(x)

r = int(y)

t= int(z)

print("The simple interst is: ", interest(p, r, t))

Q79. Write a Python program to check if a number is prime or not.

A79. num = int(input("Enter the number:\n"))

if num > 1:

for i in range(2, int(num/2)+1):

if (num % i) == 0:

print(num, "is not a prime number")

break

else:

print(num, "is a prime number")

else:

print(num, "is not a prime number")

Q80. Write a Python program to check Armstrong Number.

A80. n = int(input("Enter a number: "))

sum = 0

num = n

while num > 0:

digit = num % 10

sum += digit \*\* 4

num //= 10

if n == sum:

print(n, "is an Armstrong number")

else:

print(n, "is not an Armstrong number")

Q81. Write a Python program to find the n-th Fibonacci Number.

A81. n = int(input("Enter the n-th Fibonacci Number: "))

sum = 0

def fibo(n):

if n == 0:

return 0

else:

for i in range(1, n+1):

global sum

sum = sum + i

return sum

if n < 0:

print("Incorrect input")

else:

print("The n-th Fibonacci Number is ", fibo(n))

Q82. Write a Python program to interchange the first and last element in a list.

A82. l = []

n = int(input("Enter the number of elements: "))

for i in range(0, n):

ele = int(input())

l.append(ele)

print("The list is:", l)

l\_new = l

def interchange(l):

a = l[0]

b = l[len(l)-1]

l\_new[0] = b

l\_new[len(l)-1] = a

return l\_new

if l == []:

print("List is empty")

elif n == 1:

print("List has only one element")

else:

interchange(l)

print("The list with first and last element interchanged is ", l\_new)

Q83. Write a Python program to swap two elements in a list.

A83. l = []

n = int(input("Enter the number of elements: "))

for i in range(0, n):

ele = int(input())

l.append(ele)

print("The list is:", l)

pos1, pos2 = input("Enter the positions to be swapped: ").split()

def swapping(l, a, b):

l[a], l[b] = l[b], l[a]

return l

if l == []:

print("List is empty")

elif n == 1:

print("List has only one element")

elif int(pos1) > n or int(pos2) > n:

print("Position out of list range")

else:

swapping(l, int(pos1), int(pos2))

print("The list with first and last element interchanged is ", l)

Q84. Write a Python program to find N largest element from a list.

A85. l= []

n = int(input("Enter the number of elements: "))

for i in range(0, n):

ele = int(input())

l.append(ele)

num = int(input("Enter the number of largest elements: "))

def largest(l):

l\_ele = []

for i in range(0, num):

lar = 0

for j in range(len(l)):

if l[j] > lar:

lar = l[j]

l.remove(lar)

l\_ele.append(lar)

return l\_ele

if l == []:

print("List is empty")

elif n == 1:

print("List has only one element")

elif num > n:

print("List size is smaller than largest elements list size")

else:

print("The N largest elemts in the list is ", largest(l))

Q85. Write a Python program to find cumulative sum of a list.

A85. l = []

n = int(input("Enter the number of elements: "))

for i in range(0, n):

ele = int(input())

l.append(ele)

def cumulative(l):

sum = 0

l\_new = []

for i in range(0, n):

sum = sum + l[i]

i = i + 1

l\_new.append(sum)

return l\_new

if l == []:

print("List is empty")

else:

print("The cumulative sum of the list is ", cumulative(l))

Q86. Write a Python program to check if a string is palindrome or not.

A86. = input("Enter a string for palindrome check: ")

def pali(l):

return l == l[::-1]

if pali(s):

print("The string is a palindrome")

else:

print("The string is not a palindrome")

Q87. Write a Python program to remove i'th element from a string.

A87. s = input("Enter a string: ")

n = int(input("Enter the ith position for element removal from the string: "))

def ele\_remove(l, n):

a = s[:n]

b = s[n+1:]

return a + b

print("The string with i'th elemnent removed from the string is ", ele\_remove(s, n))

Q88. Write a Python program to check if a substring is present in a given string.

A88. s1 = input("Enter a string: ")

s2 = input("Enter the sub-string: ")

if s2 in s1:

print("The sub-string is present in the string")

else:

print("The sub-string is not present in the string")

Q89. Write a Python program to find words which are greater than given length k.

A89. s1 = input("Enter a string: ")

k = int(input("Enter the length k: "))

s = []

words = s1.split(" ")

for word in words:

if len(word) > k:

s.append(word)

print("Words that are greater than the given length k is/are: ", s)

Q90. Write a Python program to extract unquire dictionary values.

A90. t = {'a': [45, 34, 73, 232],

'b': [56, 6, 22, 10],

'c': [45, 22, 7],

'd': [73, 45, 542]}

print("The original dictionary is : " + str(t))

l = list(sorted({x for i in t.values() for x in i}))

print("The unique values list is : " + str(l))

Q91. Write a Python program to merge two dictionary.

A91. a = {1: "apple", 2: "pineapple"}

b = {1: "orange", 3: "peach"}

c = b.copy()

c.update(a)

print(c)

Q92. Write a Python program to convert a list of tuples into dictionary.

Input : [('Sachin', 10), ('MSD', 7), ('Kohli', 18), ('Rohit', 45)]

Output : {'Sachin': 10, 'MSD': 7, 'Kohli': 18, 'Rohit': 45}

A92. def tuple\_to\_dict(inp, di):

for a, b in inp:

di.setdefault(a, []).append(b)

return di

inp = [('Sachin', 10), ('MSD', 7), ('Kohli', 18), ('Rohit', 45)]

di = {}

print(tuple\_to\_dict(inp, di))

Q93. Write a Python program to create a list of tuples from given list having number and its cube in each tuple.

Input: list = [9, 5, 6]

Output: [(9, 729), (5, 125), (6, 216)]

A93. def list\_tup(l):

l2 = []

for i in range(len(l)):

a = l[i]\*\*3

b = (l[i], a)

l2.append(b)

i = i + 1

return l2

l1 = [9, 5, 6]

print(list\_tup(l1))

Q94. Write a Python program to get all combinations of 2 tuples.

Input : test\_tuple1 = (7, 2), test\_tuple2 = (7, 8)

Output : [(7, 7), (7, 8), (2, 7), (2, 8), (7, 7), (7, 2), (8, 7), (8, 2)]

A94. def combo\_tup(t1, t2):

c = [(a, b) for a in t1 for b in t2]

c = c + [(a, b) for a in t2 for b in t1]

return c

test\_tuple1 = (7, 2)

test\_tuple2 = (7, 8)

print(combo\_tup(test\_tuple1, test\_tuple2))

Q95. Write a Python program to sort a list of tuples by second item.

Input : [('for', 24), ('Geeks', 8), ('Geeks', 30)]

Output : [('Geeks', 8), ('for', 24), ('Geeks', 30)]

A95. def sort\_tup(t):

l = len(t)

for i in range(l):

for j in range(0, l-i-1):

if (t[j][1] > t[j + 1][1]):

t\_new= t[j]

t[j] = t[j + 1]

t[j + 1] = t\_new

return t

# Driver Code

t = [('for', 24), ('Geeks', 8), ('Geeks', 30)]

print(sort\_tup(t))

Q96. Write a python program to print below pattern.

\*

\* \*

\* \* \*

\* \* \* \*

\* \* \* \* \*

A96. for i in range(6):

for j in range(i):

print("\* ", end="")

print("\r")

Q97. Write a python program to print below pattern.

\*

\*\*

\*\*\*

\*\*\*\*

\*\*\*\*\*

A97. k = 2\*6 - 2

for i in range(0, 6):

for j in range(0, k):

print(end=" ")

k = k - 2

for j in range(0, i+1):

print("\* ", end="")

print("\r")

Q98. Write a python program to print below pattern.

\*

\* \*

\* \* \*

\* \* \* \*

\* \* \* \* \*

A98. k = 6 - 1

for i in range(0, 6):

for j in range(0, k):

print(end=" ")

k = k - 1

for j in range(0, i+1):

print("\* ", end="")

print("\r")

Q99. Write a python program to print below pattern.

1

1 2

1 2 3

1 2 3 4

1 2 3 4 5

A99. for i in range(6):

for j in range(i):

print(j+1, " ", end="")

print("\r")

Q100. Write a python program to print below pattern.

A

B B

C C C

D D D D

E E E E E

A100. l = ["A", "B", "C", "D", "E"]

for i in range(6):

for j in range(i):

print(l[i-1], " ", end="")

print("\r")