Python Mega Assignment

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

A1) Python is considered as High-level programming language which means it can be understood by humans and is easily interpreted. High-level languages are closer to English. Python's syntax is designed to be easy to read and understand resulting in fewer coding steps for developers as compared to other languages. Python due to its simplicity is used in various application such as data science, web and software development, automation etc and hence its called as general purpose language.

Q2) Why Python is called dynamically typed language?

A2) In other languages such as Java we have to declare the datatype of variables before using them or assigning them any value. On the other hand we do not have to declare the datatype of variable in Python, it decides the datatype at runtime. Hence, Python is dynamically typed language.

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

A3) Pros:

a) Simple and easy to learn
b) Easy to code
c) Object-Oriented
d) Extensive libraries
e) Automatic memory management

Cons:

a) Not suitable for mobile
b) Since Python is dynamic, shows more error at runtime
c) Python is slow
d) Has limitations with database access
e) Python consumes more memory

Q4) In what all domains we can use Python?

A4)

a) Data Science
b) Automation
c) Application Development
d) Artificial Intelligence and Machine Learning
e) Desktop GUI
f) Game Development

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

A5) Variables, in simple terms we can say that they are memory locations. Variables are containers for storing data values.

Python has no command for declaring a variable. A variable is created when we assign a value to it.

Eg: 1. x = 10 (type is integer)

2. y = “Data” (type is String)

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

A6) We can take input from the user using **input()** function.

Eg: name = input(“Enter name =”)

print(“Name is ”, name)

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

A7) String is the default datatype of the value that has been taken input using input() function.

Q8) What is type casting?

A8) The conversion of one datatype to another datatype is called as type casting.

Eg:

```
x = 10

y = float (x)

print (“The casted value is:”, y)
```

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

A9) Yes, we can take more than one input from the user using single input() function. We can do it by using split() method.

Eg:

```
a, b, c = input("Enter three values: ").split()
print("Enter Your First Name: ", a)
print("Enter Your Last Name: ", b)
print("Enter Your Class: ", c)
```

Q10) What are keywords?

A10) Python has a set of keywords that are reserved words that cannot be used as variable names, function names, or any other identifiers. Eg: True, False, and, if, for, else etc.

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

A11) No, we cannot use keywords as variable because keywords are predefined and have specific functions assigned to them. If we forcefully use keywords for variable it will throw an error.

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

A12) Indentation refers to the spaces that are used in the beginning of a statement. By default python puts 4 spaces but it can be changed accordingly. Eg:

```
if(condition):
    Statement
    If (condition):
        Statement 1
        Statement 2
```

Q13) How can we throw some output in Python?

A13) We can thro output using print() function.

Eg:

```
Print("Hello World")
```

Q14) What are operators in Python?

A14) Operators can be defined as a symbol which is responsible for a particular operation between two operands.

a) Numerical Operators in Python

(+) for addition

(-) for substraction

(*) for multiplication
(/) for float division
(//) for integer division
(**) for power calculation
(%) for modulus

```
x = 3
y = 5
print("Addition of x + y = ", x+y)
print("Substraction of x - y = ", x-y)
print("Multiplication of x * y = ", x*y)
print("Float Division of x / y = ", x/y)
print("Integer Divison of x // y = ", x//y)
print("Modulus of x % y = ", x%y)
print("Power of y on x i.e; x ** y = ", x**y)
```

b) Concat operator for string

```
name = “James”+ “ ” + “Bond”

print (“Full Name = ”, name)
```

c) Assignment operators

= , x = 5
+= , x += 5 -> x = x + 5
-= , x -= 5 -> x = x - 5
*= , x *= 5 -> x = x * 5
/= , x /= 5 -> x = x / 5
//= , x //= 5 -> x = x // 5
%= , x %= 5 -> x = x % 5

d) Comparison Operator

==, Equals to condition. x==y

!= , Not Equals to condition , x != y
> , Greater than condition , x > y
< , Less than condition , x < y
>= , Greater than and Equals to condition , x >= y

<= , Less than and Equals to condition , x <= y

e) Logical Operators – Logical operators in Python will check for expression results.

and -> Returns true if both statements are true
or -> Returns true if one of statements are true
not -> Reverse the result, returns false if the result is true

```
m = 10
n = 8
print("m>10 and n<10 Result " , m>10 and n<10) # False and True -> False
```

```
print("m>20 or n<10 Result " , m>10 or n<10) # False or True
->
print("not(m>20 and n<10) Result " , not(m>10 and n<10))
# not(False and True) -> not(False) -> True
```

Q15) What is the difference between / and // operators?

A15) / operator is for float division

// operator is for integer division

Q16) Write a code to give the following output

iNeuroniNeuroniNeuroniNeuron

A16) name = “iNeuron”*4

print(“Result is:”, name)

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

A17) number = input (“Enter any number”)

if (number%2)==0:

```
print(“The entered number is even !!”)


else:
print (“The entered number is odd !!”)
```

Q18) What are Boolean Operators?

A18) Boolean operators gives output as True and False depending on the condition. They only have 2 output.

For example, 1==1 is True whereas 2<1 is False.

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) 1 (True)

0 (False)

False

1 (True)

Q20) What are conditional statements in Python?

A20) Conditional statements in Python are used to handle conditions.

Eg:

If- else

```
x = 10
Y = 5
if x==y:
   print("Yes, X is Equals to Y !!")
else:
   print("No, X is not Equals to Y !!")
```

Nested if-else

```
Marks = 54

If marks >=90:
print(“Grade A+”)

elif marks >=80 and marks <90:
print(“Grade A”)

elif marks >=70 and marks <80:
print(“Grade B+”)

elif marks >=60 and marks <70:
print(“Grade B”)

else:
print(“Grade C”)
```

Q21) What is use of ‘if’, ‘elif’ and ‘else’ keywords?

A21) if, elif and else, all are conditional statements which are used in a program when we have multiple conditions and accordingly we can use them.

Eg:

If- else

```
x = 10
Y = 5
if x==y:
   print("Yes, X is Equals to Y !!")
else:
   print("No, X is not Equals to Y !!")
```

Nested if-else

```
Marks = 54

If marks >=90:
print(“Grade A+”)

elif marks >=80 and marks <90:
print(“Grade A”)

elif marks >=70 and marks <80:
print(“Grade B+”)

elif marks >=60 and marks <70:
print(“Grade B”)

else:
print(“Grade C”)
```

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 = input(“Enter your age”)

if age >18:

print (“You can vote”)

else:
print(“You 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 val in numbers:
if (val%2)==0:
sum = sum+val
print(sum)

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

A24)

n1 = int(input(“Enter first number: ”))
n2 = int(input(“Enter second number: ”))

```
n3 = int(input("Enter third number: "))

if (n1>n2) and (n1>n3):
largest = n1

elif (n2>n1) and (n2>n3):
largest = n2

else:
largest = n3

print("The largest number is", largest)
```

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]
for j in numbers:
    if j>500:
     break;

    if j==0 and j<=150:
     print(j," ");
```

Q26) What is a String? How can we declare String in Python?

A26) String is a sequence data type in Python. It is a sequence of unicode characters.

Strings are created by enclosing characters inside a single quote or double quotes.eg.

Syntax:

str = ”Hello”

Q27) How can we access the string using its index?
A27) We can access the string using indexing as strings are ordered sequences of character data. The individual characters of a string can be accessed using numerical index. Eg

Str = “laptop”
S = str[0]
print(s)

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

string = "Big Data iNeuron"
desired_output = "iNeuron"

A28. string = "Big Data iNeuron"
desired_output = (string.replace('Big Data',''))
print(desired_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"
desired_output = (string.replace('Big Data',''))

```
output = desired_output [::-1]
print(output)
```

Q30. Reverse the string given in the above question.

A30. string = "Big Data iNeuron"[::-1]

```
print(string)
```

Q31. How can you delete entire string at once?

A31. We can delete entire string at once using the del function. Eg:

```
str1= “keyboard”

del str

 print (str1)          (Note: the output will be an error)
```

Q32. What is escape sequence?

A32) Escape sequences allow you to include special characters in strings. To do this, simply add a backslash (\) before the character you want to escape.

Eg.

```
s = 'Hey, what\'s up?'

print(s)
```

Q33. How can you print the below string?

'iNeuron's Big Data Course'

A33.

```
str =  'iNeuron's Big Data Course'
print(str)
```

Q34. What is a list in Python?

A34. List - Lists are used to store multiple items in a single variable. Lists are one of 4 built-in data types in Python used to store collections of data, the other 3 are Tuple, Set and Dictionary, all with different qualities and usage.

Lists are created using square brackets:

```
list1 = ["apple", "banana", "cherry"]
print(list)
```

Q35. How can you create a list in Python?

A35. Lists are created using square brackets:

```
list1 = ["apple", "banana", "mango"]
print(list)
```

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

A36. We can access the elements in a list using indexing.

```
list1 = ["apple", "banana", "mango"]

s1 = list[1]

print (s1)
```

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"]
print(lst[4][2])
```

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

A38) n = int(input("Enter the size of list : "))

numList = list(map(float, input("Enter the list numbers separated by space").strip().split()))[:n]

print("User List: ", numList)

len(numList)

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(lst)

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

A40) The primary difference between tuples and lists is that tuples are immutable as opposed to lists which are mutable. Therefore, it is possible to change a list but not a tuple. The contents of a tuple cannot change once they have been created in Python due to the immutability of tuples.

Q41. How can you create a tuple in Python?

A41) Tuples are written with round brackets.

Eg: thistuple = ("apple", "banana", "cherry")
print(thistuple)

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) Tuples are unchangeable, meaning that we cannot change, add or remove items after the tuple has been created.

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

A43) Yes, two tuples can be appended.

Eg.

```
inputTuple_1 = (12, 8, 6)

inputTuple_2 = (3, 4)

resultTuple = inputTuple_1 + inputTuple_2

print("Resultant tuple after appending inputTuple_2 to the inputTuple_1:", resultTuple)
```

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

A44)

```
my_tuple = tuple(input('Enter your words: ').split())

print(my_tuple)

print(len(my_tuple))
```

Q45. What are sets in Python?

A45) Sets are used to store multiple items in a single variable. Set is one of 4 built-in data types in Python used to store collections of data, the other 3 are List, Tuple, and Dictionary, all with different

qualities and usage. A set is a collection which is unordered, unchangeable, and unindexed.

Eg.

```
thisset = {"apple", "banana", "cherry"}
print(thisset)
```

Q46. How can you create a set?

A46) Sets are created with curly brackets.

Eg.

```
thisset = {"apple", "banana", "cherry"}
print(thisset)
```

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

A47)
```
set1 = {'g', 'e', 'k'}

    set1.add('iNeuronset')

    print("Letters are:", set1)
```

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

A48)

```
fruits = {"apple", "banana", "cherry"}
```

fruits.update([1,2,3])

print(fruits)

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

A49) We use add() method to add single value to a set. We use update() method to add sequence values to a set. Here Sequences are any iterables including list , tuple , string , dict etc

Q50. What is clear() in sets?

A50) The clear() method removes all elements in a set.

Eg. fruits = {"apple", "banana", "cherry"}

fruits.clear()

print(fruits)

Q51. What is frozen set?

A51) Python frozenset() Method creates an immutable Set object from an iterable. It is a built-in Python function. As it is a set object therefore we cannot have duplicate values in the frozenset.

Eg. nu = ()

fnum = frozenset(nu)

print("frozenset Object is : ", fnum)

Q52. How is frozen set different from set?

A52) Frozen set is just an immutable version of a Python set object. While elements of a set can be modified at any time, elements of the frozen set remain the same after creation. Due to

this, frozen sets can be used as keys in Dictionary or as elements of another set.

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

A53) The union() method returns a set that contains all items from the original set, and all items from the specified set(s).You can specify as many sets you want, separated by commas.It does not have to be a set, it can be any iterable object.If an item is present in more than one set, the result will contain only one appearance of this item.

Eg. x = {"a", "b", "c"}

```
y = {"f", "d", "a"}

z = {"c", "d", "e"}

result = x.union(y, z)

print(result)
```

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

A54) Python set intersection() method returns a new set with an element that is common to all set

The intersection of two given sets is the largest set, which contains all the elements that are common to both sets. The intersection of two given sets A and B is a set which consists of all the elements which are common to both A and B.

Eg. s1 = {1, 2, 3}

s2 = {2, 3}

print(s1.intersection(s2))

Q55. What is dictionary in Python?

A55) Dictionary in Python is a collection of keys values, used to store data values like a map, which, unlike other data types which hold only a single value as an element.

Eg. Dict = {1: 'Geeks', 2: 'For', 3: 'Geeks'}

print(Dict)

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

A56) Dictionary in Python is an unordered collection of data values, used to store data values like a map, which unlike other Data Types that hold only single value as an element, Dictionary holds key:value pair. Key-value is provided in the dictionary to make it more optimized. Each key-value pair in a Dictionary is separated by a colon :, whereas each key is separated by a 'comma'. Lists are just like the arrays, declared in other languages. Lists need not be homogeneous always which makes it a most powerful tool in Python. A single list may contain DataTypes like Integers, Strings, as well as Objects. Lists are mutable, and hence, they can be altered even after their creation.

Q57. How can we delare a dictionary in Python?

A57) Dictionary holds key:value pair. Each key-value pair in a Dictionary is separated by a colon :, whereas each key is separated by a ‘comma’.

Eg: Dict = {1: 'Maruti', 2: 'swift', 3: 'manual'}

```
print(Dict)
```

Q58. What will the output of the following?

```
var = {}
print(type(var))
```

A58) The output will be <class ‘dict’>

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

A59) We can add an element in a dictionary using new index key and assigning a value to it.

Eg:

```
vehicle = {"brand": "Ford","model": "Mustang","year": 1964}

vehicle["color"] = "yellow"

print(vehicle)
```

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

A60) vehicle = {"brand": "Ford","model": "Mustang","year": 1964}

```
vehicle["color"] = "yellow"

x = vehicle.values()

print(x)
```

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

A61)
```
students = {
   123: {'name' : 'Alice', 'age': '23'},
   321: {'name' : 'Bob', 'age': '34'}
}

name = students[123]['name']

print(name)
```

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

A62) The get() method returns the value of the item with the specified key.

```
Eg. car = {"brand": "Ford", "model": "Mustang", "year": 1964}

x = car.get("price", 15000)

print(x)
```

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

A63) The items() method returns a view object. The view object contains the key-value pairs of the dictionary, as tuples in a list.The view object will reflect any changes done to the dictionary.

Eg

```
car = {"brand": "Ford", "model": "Mustang", "year": 1964}

x = car.items()

car["year"] = 2018

print(x)
```

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

A64) The pop() method removes the element at the specified position.

eg

```
states = ['UP', 'MP', 'Bihar']

x = states.pop(1)
```

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

A65) The popitem() method removes the item that was last inserted into the dictionary. In versions before 3.7, the popitem() method removes a random item.The removed item is the return value of the popitem() method.

Eg.

```
car = {"brand": "Ford","model": "Mustang","year": 1964}

x = car.popitem()

print(x)
```

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

A66) The keys() method returns a view object. The view object contains the keys of the dictionary, as a list. The view object will reflect any changes done to the dictionary.

Eg.

```
car = {"brand": "Ford","model": "Mustang","year": 1964}

x = car.keys()

car["color"] = "white"

print(x)
```

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

A67) The values() method returns a view object. The view object contains the values of the dictionary, as a list.

The view object will reflect any changes done to the dictionary.

Eg.

```
car = {"brand": "Ford","model": "Mustang","year": 1964}

x = car.values()

car["year"] = 2018

print(x)
```

Q68. What are loops in Python?

A68)

Loops:

1. **for loop:** Used to perform repetitive operations. To be used when we know exact number of iteration.
   - **Syntax**: for i in range (1,11)
     print i
   - Here 11 is exclusive which means if we want to print till 10 we have to mention 11
   - By default the step increment value is set to 1

```
• Write a code to print numbers from
  1 to 10 using 2 steps
• Example to print odd numbers
  starting from 1
•
• for num in range(1,11,2):
•     print(num)
```

```
    Write a code to print numbers from 10 to 1

    for num in range(10,0,-1):
    print(num)
```

```
# Write a program to calculate the sum of given
list elements using for loop
# output = 25

int_list = [4,8,-2,10,5]
list_sum = 0

for num in int_list:

   list_sum = list_sum + num

print("Total sum of elements =", list_sum)
```

2. **<u>While loop:</u>** used when number of iterations are not known

```
# Write a program to print the table of 9

num = 9
counter = 1

while counter <=10:
    ans = num*counter
    print(num, "*",counter,'=', ans)
    counter = counter + 1
```

3. **Break statement:**

```
int_list = [1,5,7,8,19,13,17,3]

# Find the even value in the given list

for num in int_list:
    print("Current element of the list = ", num)
    if  num % 2 == 0:
        print("Even number in the list =", num)
        break
```

What if break is removed

```
for num in int_list:
    print("Current element of the list = ", num)
    if  num % 2 == 0:
        print("Even number in the list =", num)
```

4. Continue – Used to bring control back to the loop as per condition

```
5.for num in range (1,21):
6.    if num<10:
7.        continue
8.    print(num)
```

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

A69) Types of loops in Python:

1. While loop
2. For loop
3. Nested loop
4. Loop control statement: continue, break, pass

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

A70)

- for: Iterate a predefined number of times. This is also known as a definite iteration

  eg: for i in range(10):

  print(i)

- while: Keep on iterating until the condition is false. This is known as an indefinite iteration.

  Eg: n = 0

  while n < 10:

  print(n)

  n += 1

Q71. What is the use of continue statement?

A71) Continue statement is a loop control statement that forces to execute the next iteration of the loop while skipping the rest of the code inside the loop for the current iteration only, i.e. when the continue statement is executed in the loop, the code inside the loop following the continue statement will be skipped for the current iteration and the next iteration of the loop will begin.

Eg. for var in "malayalam":

```
    if var == "a":
        continue
    print(var)
```

Q72. What is the use of break statement?

A72) Break statement is used to bring the control out of the loop when some external condition is triggered. break statement is put inside the loop body (generally after if condition). It terminates the current loop, i.e., the loop in which it appears, and resumes execution at the next statement immediately after the end of that loop. If the break statement is inside a nested loop, the break will terminate the innermost loop.

Eg. for i in range(10):

```
    print(i)
    if i == 2:
        break
```

Q73. What is the use of pass statement?

A73) The pass statement is a null statement. But the difference between pass and comment is that comment is ignored by the interpreter whereas pass is not ignored.

Eg)

```
a = 10

b = 20


if(a<b):

  pass

else:

  print("b<a")
```

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

A74) The Python range() function returns a sequence of numbers, in a given range.

Eg.

```
for i in range(5):

    print(i, end=" ")

print()
```

Q75. How can you loop over a dictionary?

A75) You can loop through a dictionary by using 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):

```
    return 1 if (n==1 or n==0) else n * factorial(n - 1);

num = 5;

print("Factorial of",num,"is",

factorial(num))
```

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

A77) def simple_interest(p,t,r):

```
    print('The principal is', p)

    print('The time period is', t)

    print('The rate of interest is',r)

    si = (p * t * r)/100

    print('The Simple Interest is', si)

    return si

simple_interest(6,10,8)
```

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

A78) def compound_interest(principle, rate, time):

```
    Amount = principle * (pow((1 + rate / 100), time))
    CI = Amount - principle
    print("Compound interest is", CI)
compound_interest(1000, 9.25, 8)
```

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

A79)

```
num = 105
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 = 153
s = n
b = len(str(n))
sum1 = 0
while n != 0:
    r = n % 10
    sum1 = sum1+(r**b)
    n = n//10
if s == sum1:
    print("The given number", s, "is armstrong number")
else:
    print("The given number", s, "is not armstrong number")
```

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

A81)

```
def Fibonacci(n):
        if n<= 0:
                print("Incorrect input")
        elif n == 1:
                return 0
        elif n == 2:
                return 1
        else:
                return Fibonacci(n-1)+Fibonacci(n-2)
print(Fibonacci(8))
```

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

A82)

```
def swapList(newList):
    size = len(newList)
    temp = newList[0]
    newList[0] = newList[size - 1]
    newList[size - 1] = temp
    return newList
newList = [12, 35, 9, 56, 24]
print(swapList(newList))
```

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

A83)

```
def swapPositions(list, pos1, pos2):
    list[pos1], list[pos2] = list[pos2], list[pos1]
    return list
List = [1, 2, 3, 4]
pos1, pos2  = 1, 3
print(swapPositions(List, pos1-1, pos2-1))
```

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

A84)

```
def Nmaxelements(list1, N):
    final_list = []
    for i in range(0, N):
        max1 = 0
        for j in range(len(list1)):
            if list1[j] > max1:
                max1 = list1[j];
        list1.remove(max1);
        final_list.append(max1)
    print(final_list)
list1 = [10,20,30,40,50,60,70]
N = 2
Nmaxelements(list1, N)
```

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

A85)

```
def Cumulative(lists):
    cu_list = []
    length = len(lists)
    cu_list = [sum(lists[0:x:1]) for x in range(0, length+1)]
    return cu_list[1:]
lists = [13,18,75,96,56]
print (Cumulative(lists))
```

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

A86)

```
def isPalindrome(s):
    return s == s[::-1]
s = "racecar"
ans = isPalindrome(s)
if ans:
    print("Yes")
else:
    print("No")
```

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

A87)

```
test_str = "college"

new_str = test_str[:2] + test_str[3:]

print ("The string after removal of i'th character : " + new_str)
```

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

A88)

```
string = "Ram is a good boy"

substring = "good"

s = string.split()

if substring in s:

    print("yes")

else:

    print("no")
```

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

A89)

```
def string_k(k, str):
    string = []
    text = str.split(" ")
    for x in text:
        if len(x) > k:
            string.append(x)
    return string
k = 4
str ="John and Jiya studied for mathematics exam"
print(string_k(k, str))
```

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

A90)

```
dict1 = {'infosys': [5, 6, 7, 8],'cognizant': [10, 11, 7, 5],'capgemini': [6, 12, 10, 8],'wework': [1, 2, 5]}

print("The original dictionary is : " + str(dict1))

res = list(sorted({ele for val in dict1.values() for ele in val}))

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

Q91. Write a Python program to merge two dictionary.

A91)

```
dict_1 = {'John': 15, 'Rick': 10, 'Misa' : 12 }

dict_2 = {'Bonnie': 18,'Rick': 20,'Matt' : 16 }

dict_1.update(dict_2)

print('Updated dictionary:')

print(dict_1)
```

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 Convert(tup, di):
    di = dict(tup)
    return di
tups = [('Sachin', 10), ('MSD', 7), ('Kohli', 18), ('Rohit', 45)]
dictionary = {}
print (Convert(tups, dictionary))
```

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)

```
list1 = [9,5,6]
res = [(val, pow(val, 3)) for val in list1]
print(res)
```

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)

```
test_tuple1 = (7, 2)
test_tuple2 = (7, 8)
print("The tuple 1 : " + str(test_tuple1))
print("The tuple 2 : " + str(test_tuple2))
result =  [(x, y) for x in test_tuple1 for y in test_tuple2]
result = result +  [(x, y) for x in test_tuple2 for y in test_tuple1]
print("The resultant tuple : " + str(result))
```

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_Tuple(tup):
    lst = len(tup)
    for i in range(0, lst):
        for j in range(0, lst-i-1):
            if (tup[j][1] > tup[j + 1][1]):
                temp = tup[j]
                tup[j]= tup[j + 1]
                tup[j + 1]= temp
    return tup
tup =[('for', 24), ('Geeks', 8), ('Geeks', 30)]
print(Sort_Tuple(tup))
```

Q96. Write a python program to print below pattern.

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

A96)

```
def pypart(n):
    for i in range(0, n):
      for j in range(0, i+1):
            print("* ",end="")

      print("\r")

n = 5
pypart(n)
```

Q97. Write a python program to print below pattern.

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

A97)

```
def triangle(n):
    k = n - 1
    for i in range(0, n):

        for j in range(0, k):
            print(end=" ")
        k = k - 1
        for j in range(0, i+1):
            print("* ", end="")
        print("\r")
n = 5
triangle(n)
```

Q98. Write a python program to print below pattern.

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

A98)
```
def triangle(n):
    k = n - 1
    for i in range(0, n):
        for j in range(0, k):
            print(end=" ")
        k = k - 1
        for j in range(0, i+1):
            print("* ", end="")
        print("\r")
n = 5
triangle(n)
```

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)

```
def numpat(n):
    num = 1
    for i in range(0, n):
        num = 1

        for j in range(0, i+1):
            print(num, end=" ")
            num = num + 1
        print("\r")
n = 5
numpat(n)
```

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)

```
def alphapat(n):
    num = 65
    for i in range(0, n):
        for j in range(0, i+1):
            ch = chr(num)
            print(ch, end=" ")
        num = num + 1
        print("\r")
n = 5
alphapat(n)
```