## **Assignment Part-1**

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

Ans) Python is a general-purpose programming language, because it is used in data science, software development, automation, web development and variety of applications. Python is a high-level programming language because of its ease of readability and being an interpreted language, it is not associated with the processor (Like other programming languages while compiled converts into assembly language and then runs in the processor).

Q2. Why is Python called a dynamically typed language?

Ans) Python takes care of the memory management while running the programming and it doesn't bother of, we don't give data type to any kind of variable because it changed the type of the variable during runtime of the program. So, Python is called as a dynamically typed language.

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

Ans) Pros:

1. Flexible
2. Scalable
3. Portable
4. Extensible
5. IOT opportunities
6. Embeddable
7. Larger community
8. Extensive libraries

Cons:

1. Security
2. Complex multithreading
3. Large memory wastage during garbage collection
4. Dynamically typed language
5. Design issues
6. Slower than compiled language
7. Work environment
8. High memory consumption

Q4. In what all domains can we use Python?

Ans)

1. Data Science
2. Web Development
3. Gaming
4. Scientific Programming
5. OS Development
6. Artificial Intelligence
7. Machine Learning

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

Ans) Variables are the basic unit of storage. Python has no command in declaring variable because data type of the variable is not necessary. So, declaring a variable in python is very simple.

Declaration of variable:

1. Name the variable

2. Assign value to the variable

3. It is not mandatory to declare data type of the variable, so we don't need to define it explicitly. It will be automatically assigned during the runtime of the program.

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

Ans) We can take an input from the user in Python with the help of input () function.

    Eg: a = input ("Enter input value: ")

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

Ans) String

Q8. What is type casting?

Ans) Type casting is the process of converting a data type into another data type by defining the data type as required by the user operations.

    Eg: a = 153 # which is considered as an int

        b = "23" # which is string

        c = a + b # 153+"23"

        print(c) # which throws a type conversion from string to int is not possible error.

        c = a + int(b) # where int(b) means type casting from string value to int value

        print(c) # results output as 176

    There are two types of type casting in python:

    1. Explicit type casting

    2. Implicit type casting

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

Ans) We can take more than on input from the user by using input() function with the help of split() function.

Syntax: input ().split(separator, maxsplit)

           a, b, c, d = input ("Enter four values: ").split ()

Q10. What are keywords?

Ans) Python has 33 special keywords. Keywords are the reserved words which are used to determine for only those purpose and you can't use those keywords for any other purpose rather than those special purpose. Keywords are always available and no need to import those into the code.

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

Ans) Keywords are not used as a variable because keywords are reserved, predefined and those have special meaning. keywords can be used to describe the syntax in python.

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

Ans) Python uses indentation to indicate a block of code. In other programming languages indentation is not so required and mandatory. But in Python indentation is important. Indentation is the spaces used before the line of code. Without proper indentation program returns an error.

Q13. How can we throw some output in Python?

Ans) Python can throw a custom exception by using error handling. You can choose to throw an exception if a condition arises by using the throw () keyword. In a general way we can print output by using the print () function.

Q14. What are operators in Python?

Ans) Operators are the special symbols which are used to perform some sort of computation/calculation.

Python divides the operators in the following groups:

* Arithmetic operators (+, -, \*, /, //, \*\*, etc.,)
* Comparison operators (==, ===, <, >, !=, <=, >=)
* Bitwise operators (AND, OR, NOT XOR)
* Logical operators (and, or, not)
* Identity operators (is, is not)
* Assignment operators (+=, -=, \*=, /=, \*\*=)
* Membership operators (in, not in)

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

Ans) /🡪 Float Divison

//🡪 Integer Division

Eg: x = 6

y = 3

print ('Float division of x/y= ', x/y) # Result: Float division of x/y= 2.0

print ('Integer division of x//y= ', x//y) # Result: Integer division of x//y= 2

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

iNeuroniNeuroniNeuroniNeuron

Ans) x = 'iNeuron'

y = x\*4

print(y) # Result : iNeuroniNeuroniNeuroniNeuron

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

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

if(num%2==0):

    print(“Even")

else:

    print("Odd")

Q18. What are boolean operator?

Ans) Boolean operators return True and False as an output. The logical operators **and, or and not** are also referred to as Boolean operators. And, or required two conditions to check and return the value, where not requires only one condition to be checked.

Q19. What will the output of the following?

1 or 0

0 and 0

True and False and True

1 or 0 or 0

Ans)

1 or 0 :- Output= 1

0 and 0:- Output =0

True and False and True:- Output= False

1 or 0 or 0:- Output= 1

Q20. What are conditional statements in Python?

Ans) Conditional statement is used to handle the conditions in the program. The statements in the program guide the program while making decisions based on the conditions encountered in the program. Python has 3 conditional statements:

If statement, if-else statement, and else statement.

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

Ans)

1. Python uses the ‘if’ condition to implement the decision control in the condition.

2. Along with the ‘if’ statement, ‘else’ condition can be optionally used to define an alternate block of statements to be executed if the Boolean expression in the ‘if’ condition evaluates to ‘False’.

3. ‘elif’ condition is used to include multiple conditional expressions after the ‘if’ and in between the ‘else’ if the conditions are ‘False’.

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".

Ans)

# if age >= 18 display "I can vote"

age = int (input ("Enter age: "))

if age>=18:

    print ("I can vote")

# If age is < 18 display "I can't 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]

Ans)

a=0

total=0

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

for numbers[a] in numbers:

    if(numbers[a]%2 == 0):

        total += numbers[a]

        a += 1

print(total)

#Result: 392

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

Ans)

a=int (input ("Enter num1: "))

b=int (input ("Enter num2: "))

c=int (input ("Enter num3: "))

if(a>b and a>c):

    print ("Largest number is: ", a)

elif(b>a and b>c):

    print ("Largest number is: ", b)

else:

    print ("Largest number is: ", c)

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]

Ans)

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

a = []

for i in numbers:

    if i>150:

        if i>500:

            break

        continue

    if i%5==0:

        a.append(i)

print(a)

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

Ans) String is a collection of alphabets, words, or other characters. It is one of the primitive data structures and are the building blocks for data manipulation. Python has a built-in string class named str. Python strings are "immutable" which means they cannot be changed after they are created.

Declaration of string in python:

str1 = "iNeuron"

print(str1)

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

Ans) Individual characters in a string can be accessed by specifying the string name followed by a number in square brackets ( [] ). String indexing in Python is zero-based: the first character in the string has index 0 , the next has index 1 , and so on.

Syntax:

Str1 = "iNeuron"

print(str1[4])

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

string = "Big Data iNeuron"

desired\_output = "iNeuron"

Ans) string = "Big Data iNeuron"

#desired\_output = "iNeuron"

print(string[9:16])

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

string = "Big Data iNeuron"

desired\_output = "norueNi"

Ans) string1 = "Big Data iNeuron"

#desired\_output = "norueNi"

output=string1[9:16]

print(output[::-1])

Q30. Resverse the string given in the above question.

Ans) string2 = "Big Data iNeuron"

print(string2[::-1])

Q31. How can you delete entire string at once?

Ans) We can remove entire string at once using del command.

Eg:

string\_welcome = "Hello, how are you?"

print(string\_welcome)

del(string\_welcome)

print(string\_welcome) #NameError: name 'string\_welcome' is not defined

Q32. What is escape sequence?

Ans) An escape sequence is a sequence of characters that, when used inside a character or string, does not represent itself but is converted into another character or series of characters.

Eg:

print('Interview\nBit')

Output:

Interview

Bit

Q33. How can you print the below string?

'iNeuron's Big Data Course'

Ans)

print('iNeuron\'s Big Data Course')

#Output: iNeuron's Big Data Course

Q34. What is a list in Python?

Ans) 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.

Q35. How can you create a list in Python?

Ans) In Python, a list is created by placing elements inside square brackets [] , separated by commas.

Syntax:

list1 = [2, 12.5, "Abhivarshini", 5]

print(list1)

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

Ans) The syntax for accessing the elements of a list is the same as the syntax for accessing the characters of a string. We use the index operator ( [] – not to be confused with an empty list). The expression inside the brackets specifies the index.

Syntax:

list1 = [2, 12.5, "Abhivarshini", 5]

print(list1[2])

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

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

Ans)

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

output=lst[4]

print(output[2])

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

Ans)

lst1 = [1,2,3,"Abhi", 12.5, "Varshu"]

print("Length of the list is:", len(lst1))

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

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

Ans)

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

lst.insert(2,"Big")

print(lst)

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

Ans) 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?

Ans) t2 = (20, 30, 40, 50)

print(t2)

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

Ans)

t2 = (20, 30, 40, 50,"Hi")

print(t2)

t2.append(2,"Abhivarshini") #AttributeError: 'tuple' object has no attribute 'append'

I am not able to do it because tuple is immutable in python we cannot add or modify anything in the tuple once it is already created.

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

Ans) Yes, two tuples can be appended by using the + operator.

t1=(2,3) #empty tuple

#failed assignment and append

#print(t1.append(5)) #AttributeError: 'tuple' object has no attribute 'append' because tuples are immutable

#t1[0] = 3

t2 = (20, 30, 40, 50,"Hi")

print(t1+t2)

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

Ans)

t3 = (1,2,2.5,"iNeuron", 8)

print(len(t3))

Q45. What are sets in Python?

Ans) 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. Sets only return unique values in the given data.

Q46. How can you create a set?

Ans)

set1 = {1,2,3,4,5,6,1,5}

print(set1)

#Output: {1, 2, 3, 4, 5, 6}

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

Ans)

set4 = {1,2,3,4,5,6,1,5}

print(set4)

set4.add("iNeuron")

print(set4)

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

Ans)

set4 = {1,2,3,4,5,6,1,5}

print(set4)

set4.add("Welcome","to")

print(set4)

#TypeError: add() takes exactly one argument (2 given)

It is not possible to add 2 arguments at a time in sets using add() function. We can use update() function to add more than one value into a set.

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

Ans) By using add() we can add only one argument. Update() is used to add multiple arguments at a time.

set4 = {1,2,3,4,5,6,1,5}

print(set4)

set4.update(["Welcome","to"])

print(set4)

Q50. What is clear() in sets?

Ans) Set.clear() is used to clear all the elements in the set.

p = {2, 3, 5, 7}

# clear all elements

print(p)

p.clear()

print(p)

Q51. What is frozen set?

Ans) 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.

Q52. How is frozen set different from set?

Ans) A set is a collection of items where each item is unique. This is derived from the mathematical concept of the same name. Python has two built-in types for sets: set and frozenset . A set is a mutable object while frozenset provides an immutable implementation.

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

Ans) 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. Union() returns all the unique values in the given sets.

Eg:

set\_a = {1,2,3,4,5,6}

set\_b = {3,6,8,9,10}

#Union operation

print("Union of two sets are: ", set\_a | set\_b)

print("Union of two sets are: ", set\_a.union(set\_b)) #another way of union by using union() function

#Output: Union of two sets are:  {1, 2, 3, 4, 5, 6, 8, 9, 10}

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

Ans) The intersection() method returns a set that contains the similarity between two or more sets. Meaning: The returned set contains only items that exist in both sets, or in all sets if the comparison is done with more than two sets.

Eg:

set\_a = {1,2,3,4,5,6}

set\_b = {3,6,8,9,10}

#Intersection operation

print("Intersection of two sets are: ", set\_a & set\_b)

print("Intersection of two sets are: ", set\_a.intersection(set\_b))#another way of intersection by using intersection() function

#Output: Intersection of two sets are:  {3, 6}

Q55. What is dictionary in Python?

Ans) Dictionaries are used to store data values in key:value pairs. A dictionary is a collection which is ordered\*, changeable and do not allow duplicates.

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

Ans) A dictionary is a set of key-value pairs, with the keys being unique within the dictionary. This makes the dictionary useful for storing and retrieving values using that unique key. So this kind of uniqueness make dictionary different from all other data structures.

Q57. How can we declare a dictionary in Python?

Ans) A dictionary in Python is made up of key-value pairs. In the two sections that follow you will see two ways of creating a dictionary. The first way is by using a set of curly braces, {} , and the second way is by using the built-in dict() function.

Eg:

dict2 = {}

dict2['name'] = 'Varshi'

dict2['age'] = 23

dict2['Hobby'] = 'Dancing'

dict2['skills'] = ['Python', 'Java']

dict2['States visited'] = ('AP', 'TS', 'KN')

dict2[564]='Random key'

print(dict2)

Q58. What will the output of the following?

var = {}

print(type(var))

Ans) <class ‘dict’>

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

Ans) We add a new element to the dictionary by using a new key as a subscript and assigning it a value.

Eg:

dict2['age'] = 33

print(dict2)

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

Ans) The values() method will return a list of all the values in the dictionary.

values\_total = list(dict2.values())

print("Total values in dict are: ", values\_total)

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

Ans)

people = {1: {'name': 'John', 'age': '27', 'sex': 'Male'},

          2: {'name': 'Marie', 'age': '22', 'sex': 'Female'}}

print(people[1]['name'])

print(people[1]['age'])

print(people[1]['sex'])

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

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

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

Ans) The items() method returns a view object. The view object contains the key-value pairs of the dictionary, as tuples in a list.

Syntax:

car = {

  "brand": "Ford",

  "model": "Mustang",

  "year": 1964

}

x = car.items()

print("Items in the dictionary are: ",x)

#Output: Items in the dictionary are:  dict\_items([('brand', 'Ford'), ('model', 'Mustang'), ('year', 1964)])

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

Ans) **pop()**is an inbuilt function in Python that removes and returns the last value from the [List](https://www.geeksforgeeks.org/python-list/) or the given index value.

Eg:

l = [1, 2, 3, 4]

print("Popped element:", l.pop())

print("List after pop():", l)

#Output: List after pop(): [1, 2, 3]

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

Ans) The popitem() method removes the item that was last inserted into the dictionary.

Eg:

car = {

  "brand": "Ford",

  "model": "Mustang",

  "year": 1964

}

car.popitem()

print(car)

#Output: {'brand': 'Ford', 'model': 'Mustang'}

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

Ans) The keys() method in Python Dictionary, returns a view object that displays a list of all the keys in the dictionary in order of insertion using Python.

Eg:

keys\_total = list(dict2.keys())

print("Total keys in dict are: ", keys\_total)

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

Ans) values() is an inbuilt method in Python programming language that returns a view object. The view object contains the values of the dictionary, as a list.

Eg:

values\_total = list(dict2.values())

print("Total values in dict are: ", values\_total)

Q68. What are loops in Python?

Ans) Looping means repeating something over and over until a particular condition is satisfied. A for loop in Python is a control flow statement that is used to repeatedly execute a group of statements as long as the condition is satisfied. Such a type of statement is also known as an iterative statement.

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

Ans) There are two types of loops in Python, for and while.

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

Ans) The for and while loops are both conditional statements. A **for** loop is a single-line command that will be executed repeatedly. **While** loops can be single-lined or contain multiple commands for a single condition.

Eg: To print 8 table using while loop.

i=1

while i<=10:

    ans=i\*9

    print("9 \* ",i,"= ", ans)

    i+=1

Eg: To print odd numbers from 1 to 10 using for loop.

for i in range(1,11,2):

    print(i)

Q71. What is the use of continue statement?

Ans) A continue statement ends the current iteration of a loop. Program control is passed from the continue statement to the end of the loop body. A continue statement can only appear within the body of an iterative statement, such as do , for , or while .

Q72. What is the use of break statement?

Ans) 'Break' in Python is a loop control statement. It is used to control the sequence of the loop. Suppose you want to terminate a loop and skip to the next code after the loop; break will help you do that. A typical scenario of using the Break in Python is when an external condition triggers the loop's termination.

Q73. What is the use of pass statement?

Ans) The pass statement is used as a placeholder for future code. When the pass statement is executed, nothing happens, but you avoid getting an error when empty code is not allowed. Empty code is not allowed in loops, function definitions, class definitions, or in if statements.

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

Ans) The range() function returns a sequence of numbers, starting from 0 by default, and increments by 1 (by default), and stops before a specified number.

Eg:

for i in range(1,21):

    if i<10:

        continue

    print(i)

Q75. How can you loop over a dictionary?

Ans) 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.

Eg:

for k,v in dict2.items():

    print("Key is: ",k, "and value is: ",v)

**Coding problems**

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

Ans) def factorial(n):

    if n==0:

        return 1

    elif n==1:

        return 1

    else:

        return n\*factorial(n-1)

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

print(factorial(n))

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

Ans)

P = input("Enter prinicipal amount: ")

T = input("Enter time: ")

R = input("Enter rate: ")

SI = (P\*T\*R)/100

print("Simple interest is: ", SI)

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

Ans)

def compound\_interest(principle, rate, time):

    Amount = principle \* (pow((1 + rate / 100), time))

    CI = Amount - principle

    print("Compound interest is", CI)

compound\_interest(10000, 10.25, 5)

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

Ans)

def PrimeChecker(a):

    if a > 1:

        for j in range(2, int(a/2) + 1):

            if (a % j) == 0:

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

                break

        else:

            print(a, "is a prime number")

    else:

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

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

PrimeChecker(a)

Q80. Write a Python program to check Armstrong Number.

Ans)

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

sum = 0

temp = num

while temp > 0:

   digit = temp % 10

   sum += digit \*\* 3

   temp //= 10

if num == sum:

   print(num,"is an Armstrong number")

else:

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

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

Ans)

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)

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

print(Fibonacci(n))

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

Ans)

def swapList(newList):

    size = len(newList)

    temp = newList[0]

    newList[0] = newList[size - 1]

    newList[size - 1] = temp

    return newList

newList = [1,2,3,4]

print("Elements before swapping: ", newList)

print("Elements after swapping: ",swapList(newList))

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

Ans)

def swapPositions(list, pos1, pos2):

    list[pos1], list[pos2] = list[pos2], list[pos1]

    return list

List = [81, 67, 5, 4, 39]

pos1, pos2 = 3, 5

print(swapPositions(List, pos1-1, pos2-1))

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

Ans)

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("Elements after removing: ",final\_list)

list1 = [2, 6, 41, 85, 0, 3, 7, 6, 10]

print("Elements in the list: ", list1)

N = 4

Nmaxelements(list1, N)

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

Ans)

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 = [10, 20, 30, 40, 50]

print("Elements in the list are: ", lists)

print ("Cumulative sum of a list: ",Cumulative(lists))

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

Ans)

def isPalindrome(s):

    return s == s[::-1]

s = input("Enter string: ")

ans = isPalindrome(s)

if ans:

    print("Yes")

else:

    print("No")

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

Ans)

def remove(string, i):

    a = string[ : i]

    b = string[i + 1: ]

    return a + b

if \_\_name\_\_ == '\_\_main\_\_':

    string = "geeksFORgeeks"

    i = 11

    print(remove(string, i))

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

Ans)

MyString1 = str(input("Enter string: "))

inp = input("Enter string to search: ")

if inp in MyString1:

    print("Yes! it is present in the string")

else:

    print("No! it is not present")

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

Ans)

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 ="geek for geeks"

print(string\_k(k, str))

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

Ans)

dict1 = {'A' : [1, 3, 5, 4],

             'B' : [4, 6, 8, 10],

             'C' : [6, 12, 4 ,8, 1],

             'D' : [5, 7, 2, 8, 9]}

print("The original dictionary is : " ,dict1)

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

print("The unique values list is : " , res)

Q91. Write a Python program to merge two dictionary.

Ans)

def Merge(dict1, dict2):

    return(dict2.update(dict1))

dict1 = {'a': 10, 'b': 8}

dict2 = {'e': 12, 'd': 6, 'c': 4}

print(Merge(dict1, dict2))

print(dict2)

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}

Ans)

def Convert(tup, di):

    for a, b in tup:

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

    return di

tups = [("Sachin", 10), ("MSD", 7), ("Kohli", 18), ("Rohit", 45)]

print("Input: ", tups)

dictionary = {}

print ("Output in dictionary format: ",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)]

Ans)

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)]

Ans)

from itertools import chain, product

tuple1 = (7, 2)

tuple2 = (7, 8)

print("The tuple 1 : " + str(tuple1))

print("The tuple 2 : " + str(tuple2))

result = list(chain(product(tuple1, tuple2), product(tuple2, 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)]

Ans)

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("Input tuple is: ", tup)

print("After sorting a list of tuples by second item: ", Sort\_Tuple(tup))

Q96. Write a python program to print below pattern.

\*

\* \*

\* \* \*

\* \* \* \*

\* \* \* \* \*

Ans)

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.

\*

\*\*

\*\*\*

\*\*\*\*

\*\*\*\*\*

Ans)

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.

\*

\* \*

\* \* \*

\* \* \* \*

\* \* \* \* \*

Ans)

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

Ans)

rows = 5

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

    for j in range(1, i + 1):

        print(j, end=' ')

    print('')

Q100. Write a python program to print below pattern.

A

B B

C C C

D D D D

E E E E E

Ans)

n = 5

p=65

for i in range(n):

   for j in range(i+1):

      print(chr(p), end=' ')

   p+=1

   print()