Assignment Part-1

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

ANS - Python is called a general-purpose programming language because it is the one that’s usable (not just theoretically, but practically) for solving a very wide range of different kinds of

problems.In Python, you can write detailed numerical simulations, web microservices and so on.

Python is high-level programming language means it's easy for humans to understand because-

1- Python is easy to use:

2 -Python runs on any platform:

3 - Incredible Artificial Intelligence and Machine Learning support:

**Q2. Why is Python called a dynamically typed language?**

Python is a dynamically typed language. because We don't have to declare the type of a variable or manage the memory while assigning a value to a variable in Python.

Python don't have any problem even if we don't declare the type of variable. It states the kind of variable in the runtime of the program.

Python also take cares of the memory management which is crucial in programming. So, Python is a dynamically typed language.

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

Advantages:

1- Versatile, easy to use and fast to develop:

Python programming language focuses on code readability. It’s versatile, neat, easy to use and learn, readable, and well-structured.

2- Open source with a vibrant Python community

You can download Python for free and write asynchronous code in a matter of minutes. Developing with Python is hassle-free.

3-Python has a vast collection of libraries

4- Python is an interpreted language

5- Dynamically Typed

Disadvantages:

1. Slow Speed

2. Not Memory Efficient - The Python programming language uses a large amount of memory.

3. Runtime Errors - As we know Python is a dynamically typed language so the data type of a variable can change anytime.

A variable containing integer number may hold a string in the future, which can lead to Runtime Errors.

**Q4. In what all domains can we use Python?**

Ans- Python allows the user to work on multiple domains ranging from Data Science, Machine Learning, Deep Learning,

Artificial Intelligence, Scientific Computing Scripting, Networking, Game Development to Web Development

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

Python Variable is containers which store values. Python is not “statically typed”. We do not need to declare variables before using them or declare their type.

A variable is created the moment we first assign a value to it. A Python variable is a name given to a memory location. It is the basic unit of storage in a program.

Var = "Rahul Jain"

Var2=345

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

name = input("Enter your name:")

print("name is: " + name)

input (): This function first takes the input from the user and converts it into a string. The type of the returned object always will be <type ‘str’>.

It does not evaluate the expression it just returns the complete statement as String.

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

The default type taken by input function is string.

**Q8. What is type casting?**

Type Casting is the method to convert the variable data type into a certain data type in order to the operation required to be performed by users.

Implicit Type Conversion

In this, methods, Python converts data type into another data type automatically. In this process, users don’t have to involve in this process.

a = 7

print(type(a))

Explicit Type Conversion

Mainly in type casting can be done with these data type function:

Int() : Int() function take float or string as an argument and return int type obje ct.

float() : float() function take int or string as an argument and return float type object.

str() : str() function take float or int as an argument and return string type object.

# int variable

a = 5

# typecast to float

n = float(a)

print(n)

print(type(n))

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

Python user can take multiple values or inputs in one line by two methods.

* Using split() method

Generally, users use a split() method to split a Python string but one can use it in taking multiple inputs

Example:

# taking two inputs at a time

x, y = input("Enter two values: ").split()

print("Number of boys: ", x)

print("Number of girls: ", y)

print()

* Using List comprehension

List comprehension is an elegant way to define and create list in Python. We can create lists just like mathematical statements in one line only. It is also used in getting multiple inputs from a user

# taking two input at a time

x, y = [int(x) for x in input("Enter two values: ").split()]

print("First Number is: ", x)

print("Second Number is: ", y)

print()

.

**Q10. What are keywords?**

**Keywords** are some predefined and reserved words in python that have special meanings. Keywords are used to define the syntax of the coding. The keyword cannot be used as an identifier, function, and variable name. All the keywords in python are written in lower case except True and False. There are 33 keywords in Python 3.7

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

A keyword is a word that is reserved for use by the programming language such as ‘int’ to indicate an integer, ‘if’ when using a condition statement or ‘for’ when using a loop. As a result, these cannot be used to name a variable. A variable is a value that changes during the running of a program. Examples of this are the value of a counter, the colour of a box or the total of a set of values of ages of a group. A variable has a name and is normally set as a certain type of variable (such as int, char, string). A variable cannot have a name of a keyword used by the programming language.

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

Indentation refers to the spaces at the beginning of a code line.

Where in other programming languages the indentation in code is for readability only, the indentation in Python is very important.

Python uses indentation to indicate a block of code.

if 5 > 2:  
 print("Five is greater than two!")   
if 5 > 2:  
        print("Five is greater than two!")

Python will give you an error if you skip the indentation:

**Q13. How can we throw some output in Python?**

Python provides the [print()](https://www.geeksforgeeks.org/python-output-using-print-function/) function to display output to the standard output devices.

Ex- print("Rahul")

**Q14. What are operators in Python?**

Operators are used to perform operations on variables and values.

Python divides the operators in the following groups:

* Arithmetic operators
* Assignment operators
* Comparison operators
* Logical operators
* Identity operators
* Membership operators
* Bitwise operators

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

/ is a normal division operator and // is a floor division operator. Now the question is what is normal division operator and what is floor division operator? Normal division operator gives exact result value when we divide something however floor division operator gives the floor value of exact result value when we divide something.

For example - 5/2 will result 2.5 so here 2.5 is the exact result value and floor value of 2.5 would be 2 so 5//2 will result 2.

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

**iNeuroniNeuroniNeuroniNeuron**

**Sol- String=”iNeuron”**

**String2=String\*3**

**Print(String2)**

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

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

if n%2==0:

print(n,"is an even number")

else:

print(n,"is an odd number")

**Q18. What are boolean operator?**

The logical operators and, or and not are also referred to as boolean operators.

* *The AND operator (&& or "and")*
* *The OR operator (|| or "or")*
* *The NOT operator (not)*

Q19. What will the output of the following?

1 or 0

o/p - True

0 and 0

o/p - false

True and False and True

o/p-false

1 or 0 or 0

o/p=true

**Q20. What are conditional statements in Python?**

**Python has 3 key Conditional Statements that you should know:**

* ***if*statement**
* ***if-else* statement**
* ***if-elif-else* ladder**

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

**if statement**

If the condition following the keyword if evaluates as true, the block of code will execute

if True:

print('If block will execute!')

### elif statement

Multiple conditions can be checked by including one or more elif checks after your initial if statement. Just keep in mind that only one condition will execute:

z = 7

if z > 8:

print("I won't print!") #this statement does not execute

elif z > 5:

print("I will!") #this statement will execute

elif z > 6:

print("I also won't print!") #this statement does not execute

else:

print("Neither will I!") #this statement does not execute

### else statement

You can optionally add an else response that will execute if the condition is false:

if not True:

print('If statement will execute!')

else:

print('Else statement will execute!')

**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".**

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

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]

sol –

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

sum=0

for x in numbers:

if x%2==0:

sum=sum+x

print(sum)

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

a=int(input("ENter the first number: "))

b=int(input("ENter the second number: "))

c=int(input("ENter the third number: "))

if a>=b and a>=c:

print(a," first nunber is greatest")

elif b>=a and b>=c:

print(b,"second number is greatest")

else:

print(c,"third number is greatest")

**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]**

**sol-**

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

b = []

for i in a:

if i > 150:

if i > 500:

break

continue

if i % 5 == 0:

b.append(i)

print(b)

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

A string is a data structure in Python that represents a sequence of characters. It is an immutable data type, meaning that once you have created a string, you cannot change it.

String1 = ‘welcome to the bhopal’

print("String with the use of Single Quotes: ")

print(String1)

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

In Python indexing of strings starts from 0 till n-1, where n is the size of string. So characters in string of size n, can be accessed from 0 to n-1.

sampleStr = "Hello, this is a sample string"

print( "Character at index 5 is : " , sampleStr[5] )

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

**string = "Big Data iNeuron"**

**desired\_output = "iNeuron"**

**sol-**

string="Big Data iNeuron"

print(string[9:16])

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

**string = "Big Data iNeuron"**

**desired\_output = "norueNi"**

**sol-**

string ="Big Data iNeuron"

print(string[:7:-1])

**Q30. Resverse the string given in the above question.**

**Sol-**

string ="Big Data iNeuron"

print(string[::-1])

**Q31. How can you delete entire string at once?**

You can delete the entire string using the del command

string ="Big Data iNeuron"

del string

print(string)

**Q32. What is escape sequence?**

An escape sequence is a sequence of characters with special meaning when used inside a string or a character.

\' Prints a single quote inside a string enclosed with single quotes.

\" Prints a double quote inside a string enclosed with double-quotes.

\n Prints the succeeding part of \n even if in the same line, in a new line

\t Gives a tab space equivalent to 8 normal spaces.

\r Brings the cursor to the starting of the line, called as 'carriage return.'

\b Gives a backspace.

\f Creates an f-string which is a whole new string formatting mechanism

**Q33. How can you print the below string?**

**'iNeuron's Big Data Course'**

Sol-

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

print(string)

**Q34. What is a list in Python?**

a list is a collection of things, enclosed in [ ] and separated by commas. **Python Lists**are just like dynamically sized arrays, declared in other languages. Python lists are mutable type which implies that we may modify its element after it has been formed.

**Q35. How can you create a list in Python?**

list1 = [1, 2, "Python", "Program", 15.9]

list2 = ["Amy", "Ryan", "Henry", "Emma"]

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

**To access list elements based on the given index** – we simply pass the index starting from **0 to length-1** to access the particular element and we can also pass the negative index to access the list elements in the reverse order

# Accessing the elements of a list by its index

print("list1[0]: ", list1[0])

print("list1[1]: ", list1[1])

print("list1[2]: ", list1[2])

print("list1[3]: ", list1[3])

print("list1[4]: ", list1[4])

print() # prints a new line

# Accessing the elements of a list by its index

# in reverse order

print("list1[-1]: ", list1[-1])

print("list1[-2]: ", list1[-2])

print("list1[-3]: ", list1[-3])

print("list1[-4]: ", list1[-4])

print("list1[-5]: ", list1[-5])

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

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

sol- print(lst[4][2])

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

**Sol- lst = []**

# number of elements as input

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

# iterating till the range

for i in range(0, n):

ele = int(input())

lst.append(ele) # adding the element

print(lst)

print(len(lst))

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

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

**sol-**

lst2 = ["Welcome", "to", "Data", "course"]

lst2.insert(2,"Big")

print(lst2)

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

Sol-

|  |  |
| --- | --- |
| List are mutable | Tuples are immutable |
| 2 | Iterations are time-consuming | Iterations are comparatively Faster |
| 3 | Inserting and deleting items is easier with a list. | Accessing the elements is best accomplished with a tuple data type. |
| 4 | Lists consume more memory | Tuple consumes less than the list |
| 5 | Lists have several built-in methods. | A tuple does not have many built-in methods because of immutability |
| 6 | A unexpected change or error is more likely to occur in a list. | In a tuple, changes and errors don't usually occur because of immutability. |

**Q41. How can you create a tuple in Python?**

A tuple in Python is similar to a list. The difference between the two is that we cannot change the elements of a tuple once it is assigned whereas we can change the elements of a list.

A tuple is created by placing all the items (elements) inside parentheses (), separated by commas. The parentheses are optional, however, it is a good practice to use them.

A tuple can have any number of items and they may be of different types (integer, float, list, [string](https://www.programiz.com/python-programming/string), etc.).

# Different types of tuples

# Empty tuple

my\_tuple = ()

print(my\_tuple)

# Tuple having integers

my\_tuple = (1, 2, 3)

print(my\_tuple)

# tuple with mixed datatypes

my\_tuple = (1, "Hello", 3.4)

print(my\_tuple)

# nested tuple

my\_tuple = ("mouse", [8, 4, 6], (1, 2, 3))

print(my\_tuple)

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

**Sol-** You can't add elements to a tuple because of their immutable property. There's no append() or extend() method for tuples, You can't remove elements from a tuple, also because of their immutability.

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

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:\n", resultTuple)

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

**Sol- my\_tuple = tuple(input('Enter space-separated words: ').split())**

**count=0**

**for x in my\_tuple:**

**count+=1**

**print(count)**

**Q45. What are sets in Python?**

A Set is an unordered collection data type that is iterable, mutable, and has no duplicate elements.

*Set are represented by { } (values enclosed in curly braces)*

The major advantage of using a set, as opposed to a list, is that it has a highly optimized method for checking whether a specific element is contained in the set. This is based on a data structure known as a hash table. Since sets are unordered, we cannot access items using indexes as we do in lists.

**Q46. How can you create a set?**

myset = set(["a", "b", "c"])

print(myset)

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

**Sol –** my\_set =set()

My\_set.add(“iNeuron”)

Print(my\_set)

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

my\_set = set()

my\_set.update(["iNeuron","Rahul","Jain"])

print(my\_set)

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

To add one item to a set use the add() method.

To add more than one item to a set use the update() method**.**

**Q50. What is clear() in sets?**

Clear is the method which removes all the elements in the set

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

fruits.clear()

print(fruits)

**Q51. What is frozen set?**

**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.

nu = ()

# converting tuple to frozenset

fnum = frozenset(nu)

# printing empty frozenset object

print("frozenset Object is : ", fnum)

**Q52. How is frozen set different from set?**

Frozenset is similar to set in Python, except that frozensets are immutable, which implies that once generated, elements from the frozenset cannot be added or removed. This function accepts any iterable object as input and transforms it into an immutable object.

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

The Python set union() method returns a new set with distinct elements from all the sets.

A = {2, 3, 5}

B = {1, 3, 5}

print('A U B = ', A.union(B))

# Output: A U B = {1, 2, 3, 5}

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

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

s1 = {1, 2, 3}

s2 = {2, 3}

print(s1.intersection(s2))

o/p= {2, 3}

**Q55. What is dictionary ibn Python?**

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.

thisdict = {

"brand": "Ford",

"model": "Mustang",

"year": 1964

}

print(thisdict)

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

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.

**Q57. How can we delare a dictionary in Python?**

thisdict = {

"brand": "Ford",

"model": "Mustang",

"year": 1964

}

**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?**

thisdict = {

"brand": "Ford",

"model": "Mustang",

"year": 1964

}

thisdict["color"] = "red"

print(thisdict)

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

thisdict = {

"brand": "Ford",

"model": "Mustang",

"year": 1964

}

x = thisdict["model"]

y=thisdict[“brand”]

z=thisdict[“year”]

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

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?**

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

dictionary.get(keyname, value)

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

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, see example below.

car = {  
  "brand": "Ford",  
  "model": "Mustang",  
  "year": 1964  
}  
  
x = car.items()  
  
print(x)

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

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

**Syntax**

list.pop(pos)

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

Python dictionary popitem() method removes the last inserted key-value pair from the dictionary and returns it as a tuple.

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

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.

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

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, see example below.

dictionary.values()

**Q68. What are loops in Python?**

A loop statement allows us to execute a statement or group of statements multiple times.

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

|  |  |  |
| --- | --- | --- |
| 1 | **While loop** | Repeats a statement or group of statements while a given condition is TRUE. It tests the condition before executing the loop body. |
| 2 | **For loop** | This type of loop executes a code block multiple times and abbreviates the code that manages the loop variable. |
| 3 | **Nested loops** | We can iterate a loop inside another loop. |

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

|  |  |
| --- | --- |
| **While loop** | Repeats a statement or group of statements while a given condition is TRUE. It tests the condition before executing the loop body. |
| **For loop** | This type of loop executes a code block multiple times and abbreviates the code that manages the loop variable. |

**Q71. What is the use of continue statement?**

The continue keyword is used to end the current iteration in a for loop (or a while loop), and continues to the next iteration.

i = 0

while i < 9:

i += 1

if i == 3:

continue

print(i)

**Q72. What is the use of break statement?**

## ****Python break statement****

**break statement in**[**Python**](https://www.geeksforgeeks.org/python-programming-language/) 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.

**Q73. What is the use of pass statement?**

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.

def myfunction():  
  pass

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

The Python**range() function**returns a sequence of numbers, in a given range. The most common use of it is to iterate sequence on a sequence of numbers using [Python](https://www.geeksforgeeks.org/python-programming-language/) loops.

# using python range() function

for i in range(5):

    print(i, end=" ")

print()

**Q75. How can you loop over a dictionary?**

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.

for x in thisdict:  
  print(x)

**Coding problems**

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

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

fact=1

if(n<0):

    print("Factorial does not exist for negative numbers")

elif(n==0):

    print("Factorial is 1 for zero")

else:

   for x in range(1,n+1):

    fact=fact\*x

print(fact)

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

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

# Driver code

simple\_interest(8, 6, 8)

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

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

    # Calculates compound interest

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

    CI = Amount - principal

    print("Compound interest is", CI)

# Driver Code

compound\_interest(10000, 10.25, 5)

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

number = int(input("Enter any number: "))

if number > 1:

for i in range(2, number):

if (number % i) == 0:

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

break

else:

print(number, "is a prime number")

else:

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

**Q80. Write a Python program to check Armstrong Number.**

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

sum=0

n=len(str(a))

temp=a

while temp>0:

r=temp%10

sum+=r\*\*n

temp=temp//10

if sum==a:

print(a,"is an armstrong number")

else:

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

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

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

list=[3,5,7,9,"rahul",4,"Jain"]

n=len(list)

temp=list[0]

list[0]=list[n-1]

list[n-1]=temp

print(list)

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

**list=[3,5,7,9,"rahul",4,"Jain"]**

**n=len(list)**

**temp=list[2]**

**list[2]=list[n-3]**

**list[n-3]=temp**

**print(list)**

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

**def largest(list,n):**

**new\_list=[]**

**for x in range(0,n):**

**max=0**

**for i in range(len(list)):**

**if list[i]>max:**

**max=list[i]**

**list.remove(max)**

**new\_list.append(max)**

**return new\_list**

**list=[23,45,65,78,21,5,74,43,98]**

**n=3**

**print("Largest elements",largest(list,n))**

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

**def cummulative\_sum(list):**

**list1=[]**

**sum=0**

**for x in range(len(list)):**

**sum=sum+list[x]**

**list1.append(sum)**

**return list1**

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

**print("the cummulative sum of list",cummulative\_sum(list))**

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

def palindrome(str):

str1=str.lower()

if str1==str1[::-1]:

print("Palindrome")

else:

print("not palindrome")

palindrome("Mam")

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

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

**MyString1 = "A geek in need is a geek indeed"**

**if "need" 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.**

**str="hello my name is rahul jain"**

**k=4**

**s=str.split()**

**for x in s:**

**if len(x)>4:**

**print(x)**

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

**test\_dict = {'gfg': [5, 6, 7, 8],**

**'is': [10, 11, 7, 5],**

**'best': [6, 12, 10, 8],**

**'for': [1, 2, 5]}**

**print("The original dictionary is : " + str(test\_dict))**

**res = list(sorted({ele for val in test\_dict.values() for ele in val}))**

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

**Q91. Write a Python program to merge two dictionary.**

**def Merge(dict1, dict2):**

**return(dict2.update(dict1))**

**# Driver code**

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

**dict2 = {'d': 6, 'c': 4}**

**# This returns None**

**print(Merge(dict1, dict2))**

**# changes made in 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}**

def Convert(tup, di):

for a, b in tup:

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

return di

# Driver Code

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)]**

Sol - list = [9, 5, 6]

# using list comprehension to iterate each

# values in list and create a tuple as specified

res = [(val, pow(val, 3)) for val in list1]

# print the result

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)]**

test\_tuple1 = (7, 2)

test\_tuple2 = (7, 8)

print("The original tuple 1 : " + str(test\_tuple1))

print("The original tuple 2 : " + str(test\_tuple2))

res = [(a, b) for a in test\_tuple1 for b in test\_tuple2]

res = res + [(a, b) for a in test\_tuple2 for b in test\_tuple1]

print("The filtered tuple : " + str(res))

**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)]**

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), ('is', 10), ('Geeks', 28),

('Geeksforgeeks', 5), ('portal', 20), ('a', 15)]

print(Sort\_Tuple(tup))

**Q96. Write a python program to print below pattern.**

**\***

**\* \***

**\* \* \***

**\* \* \* \***

**\* \* \* \* \***

def pattern(n):

for i in range(0,n):

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

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

print("\r")

pattern(5)

**Q97. Write a python program to print below pattern.**

**\***

**\*\***

**\*\*\***

**\*\*\*\***

**\*\*\*\*\***

**Q98. Write a python program to print below pattern.**

**\***

**\* \***

**\* \* \***

**\* \* \* \***

**\* \* \* \* \***

def pattern(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")

pattern(5)

**Q99. Write a python program to print below pattern.**

**1**

**1 2**

**1 2 3**

**1 2 3 4**

**1 2 3 4 5**

def pattern(n):

    for i in range(0,n):

        k=1

        for j in range(0,i+1):

         print(k,end="")

         k=k+1

        print("\r")

pattern(5)

**Q100. Write a python program to print below pattern.**

**A**

**B B**

**C C C**

**D D D D**

**E E E E E**

def pattern(n):

    for i in range(0,n):

        k=65

        for j in range(0,i+1):

            ch=chr(k)

            print(ch,end=" ")

            k=k+1

        print("\r")

pattern(5)