**## PYTHON MEGA ASSIGNMENT**

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

Python is considered as general purpose language because it is not associated with a particular industry. It covers a wide range of tasks and it designed to be used in a wide range of applications such as data science, software and web development, automation and used to create a variety of different programs and isn’t specialized for any specific problems. Also, it is known as a high-level programming language as it is known for its ease of readability. Pythons’ syntax is to be read and understand resulting in fewer coding steps for developers than imposed by Java or C++

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

We don’t have to declare the type of a variable or manage the memory while assigning a value to a variable in Python whereas in other languages there is a strict declaration of variables before assigning a variable. Python states the kind of variable in the runtime of program. It also takes care of memory management which is crucial in programming. So, python is called as a dynamically typed language.

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

**PROS:**

1. It is easy to learn and read which will facilitate the way to become a top python developer
2. It has a vast collection of libraries, so one doesn’t have to depend on external libraries because it has more than enough functions which will need to carry out in project
3. It is portable programming language – where one can work on any platform without requiring the developer to make changes to the code.
4. It is free, open-source, and has a vibrant community which allows the users to access the source code and legally share the outcome of modifications.

**CONS:**

1. It has speed limitations, while your program runs in python, it has to do more work in line-by-line execution, so the process will be slow
2. It is not so strong with mobile computing. Low rate of processing a program, as well as the sub-par memory efficiency, are the two major reasons why mobile computing is not supported in python
3. Python can have run time errors because of the dynamical typing feature.
4. It consumes a lot of memory space , so if you are building an application that needs memory optimization, it will restrict your memory space.

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

In several domains python can be used such as a Machine Learning/ Artificial intelligence, Desktop GUI, Data Analytics and visualization, Big Data, Web Development, Embedded systems

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

Variable cab be defined as a name of specific memory location. To declare a variable in python just name the variable and assign the require value to it, the datatype will be automatically determined from the value assigned, we need not define it explicitly.

There are certain rules while declaring a variable such as special character such @,# are not allowed, variable name should not start with numeric values, variable can start with underscore, alphabets and it can be mix of alpha-numeric values but cannot startwith numeric values.

**Eg:** var1 = 10, \_var = 1, var\_str\_1 = “hi”

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

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

**Eg:** name = input(“enter name”)

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

Default data type is string. <class str>

**Q8. What is type casting?**

It is a method to convert the variable datatype into a required datatype in order to perform different operations by users.

**Eg:** int\_data = 10

new\_int data = float(int\_data)

print(new\_int\_data)

**O/P**: 10.0

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

Yes using split() method we can take multiple inputs from users

**Eg:** x,y = input(“Enter two values: “).split()

print(“No of boys:”, x)

print(“No of girls:”,y)

**O/P:** Enter a two value: 5 10

No of boys: 5

No of girls: 10

**Q10. What are keywords?**

Keywords are some predefined and reserved words in python that have special meanings. It is used to define syntax of the coding. The keyword cannot be used as an identifier, function, variable name.

**Eg :** and, or, not

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

The keyword cannot be used as a variable name as it will create ambiguity. Keywords have predefined meanings and used to define the syntax and structure of the python language

**Q12. What is indentation? What's the use of indentation 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, but in python it is important as it uses to indicate a block of code

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

We use the print() function to output data to the screen

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

In python operators are special symbols that designate that some sort of computation should be performed. There are different types od operators such as arithmetic operators, comparison and logical operators

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

/ - used for float division

// - used for integer division

**Eg:** x=5, y=3

print(“Float division of x/y is”, x/y)

print(“Integer division of x//y is”, x//y)

O/P:

Float division of x/y is 1.66

Integer division of x//y is is 1

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

**```**

**iNeuroniNeuroniNeuroniNeuron**

**```**

str\_data = "iNeuron"

new\_str\_data = str\_data \* 4

print(new\_str\_data)

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

number = int(input("enter number = "))

if number % 2 == 0:

print("even")

else:

print("odd")

**Q18. What are boolean operator?**

Boolean operators are those that result in the Boolean values of True and False. These include and, or, not. They are mostly used in arithmetic computations and logical comparisions.

**Q19. What will the output of the following?**

**```**

**1 or 0 🡪** True

**0 and 0 🡪** False

**True and False and True 🡪** False

**1 or 0 or 0 🡪** True

**```**

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

It is used to handle conditions in your program. These statements guide the program while making decisions based on the conditions encountered by the program.

**Eg:** if, if-else, nested if

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

The if/elif/else structure is a common way to control the flow of a program, allowing you to execute specific blocks of code depending on the value of some data.

**if statement**

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

### else statement

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

### elif statement

Multiple conditions can be checked by including one or more elif checks after your initial if statement.

**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 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]**

**```**

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

sum = 0

for i in numbers:

if i % 2 == 0:

sum = sum + i

print("Total sum of even numbers is :", sum)

**O/P:**

Total sum of even numbers is : 392

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

num1 = input("enter first number:")

num2 = input("enter second number:")

num3 = input("enter third number:")

if(num1 >= num2) and (num1 >= num3):

largest = num1

elif(num2 >= num3) and (num2 >= num1):

largest = num2

else:

largest = num3

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

**O/P:**

enter first number:89

enter second number:98

enter third number:2

The greatest number is : 98

**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]**

**```**

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

for i in numbers:

if i > 500:

break

if i % 5 == 0 and i<=150:

print(i, " ")

**O/P:**

75

150

145

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

Strings are series of characters. In python strings are immutable. String can be declared as we put sequence of characters in single or double quotes

Ex: str = “hello “

Str1 = ‘hi’

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

we can access the string by refering to its index value

eg – str = anusha

str[2] = u

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

**string = "Big Data iNeuron"**

**desired\_output = "iNeuron"**

string = "Big Data iNeuron"

print(string[9:])

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

**string = "Big Data iNeuron"**

**desired\_output = "norueNi"**

string = "Big Data iNeuron"

new = string[9:]

print(new[::-1])

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

string = "Big Data iNeuron"

print(string[::-1])

**O/P:** norueNi ataD giB

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

Using del command

str = “hi”

print(str)

del str

print(str)

**Q32. What is escape sequence?**

It is a sequence of characters with special meaning when used inside a string or character

Ex: \n, \t

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

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

str1 = "iNeuron's Big Data Course"

print(str1)

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

List is a equence datatype which is used to store heterogenous data.

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

It can be created by just placing the sequence inside the [ ] square brackets

Ex; List = [ 1,”hi”, 3.0]

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

we can access a ;ist using its index value

ex: list = [ 1,2,3]

print(list[1]) 🡪 2

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

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

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.**

list = [ ]

n = int(input("enter number of list elements:"))

for i in range(0,n):

l = str(input("enter elements"))

list.append(l)

print(list)

print(len(list))

**O/P:**

enter number of list elements:5

enter elementsapple

enter elementshelo

enter elementshi

enter elementsbye

enter elementsmoon

['apple', 'helo', 'hi', 'bye', 'moon']

5

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

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

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

lst.insert(3, 'Big')

print("finali list:",lst)

**O/P:**

finali list: ['Welcome', 'to', 'Data', 'Big', 'course']

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

Tuples are also a sequence data type containing elements of different data types.

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.

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

Here’s an example of declaring a tuple in python.

num\_tuple = (1,2,3,4,5)

print(num\_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.**

We cannot add or assign a new value to the tuple. It gives error. Below is the example how we can create empty tuple

t = ()

print(t)

t= ("anusha", "aravind")

print(t)

**O/P:**

()

('anusha', 'aravind')

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

No tuples cannot be appended as it is immutable which means that once a tuple is created you cannot delete or change the values of the items stored in it. You cannot add new values either.

Ex: t= ("anusha", "aravind")

t1 = ("hello", "bye")

t = t.append(t1)

print(t)

**O/P:**

It gives the error - AttributeError: 'tuple' object has no attribute 'append'.

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

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

print(my\_tuple)

print("Count of elemnts is",len(my\_tuple))

**O/P:**

Enter space-separated words: 5 hello 23 bye

('5', 'hello', '23', 'bye')

Count of elemnts is 4

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

Sets are used to store multiple items in a single variable.

A set is a collection which is unordered, unchangeable\*, and unindexed.

Set items are unchangeable, but you can remove items and add new items.

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

It can be created using curly braces.

**Ex:**

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

print(thisset)

**O/P:**

{'cherry', 'banana', 'apple'}

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

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

print(thisset)

thisset.add("iNeuron")

print(thisset)

**O/P:**

{'banana', 'cherry', 'apple'}

{'banana', 'cherry', 'iNeuron', 'apple'}

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

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

print(thisset)

tup = (9,8,7,2)

list = [21,22,23]

thisset.add("iNeuron")

thisset.add(1)

thisset.add(1)

thisset.add(3)

thisset.add("cococola")

thisset.add(tup) #for tup add is working, for list we should use update

thisset.update(list)

print(thisset)

**O/P:**

{'apple', 'cherry', 'banana'}

{1, 3, 'cococola', 'cherry', 'iNeuron', 21, 22, 23, 'apple', (9, 8, 7, 2), 'banana'}

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

The differences are:

Use add() function to add a single element. Whereas use update() function to add multiple elements to the set.

add() is faster than update().

add () accepts immutable parameters only. Whereas accepts iterable sequences.

add() accepts a single parameter, whereas update() can accept multiple sequences.

Ex:

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

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

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

print(fruits)

fruits.clear()

print(fruits)

**O/P:**

{'apple', 'banana', 'cherry'}

set()

**Q51. What is frozen set?**

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. But like sets, it is not ordered (the elements can be set at any index).

SYNTAX: frozenset([iterable])

EX:

# tuple of vowels

vowels = ('a', 'e', 'i', 'o', 'u')

fSet = frozenset(vowels)

print('The frozen set is:', fSet)

print('The empty frozen set is:', frozenset())

# frozensets are immutable

fSet.add('v')

**O/P:**

The frozen set is: frozenset({'a', 'o', 'u', 'i', 'e'})

The empty frozen set is: frozenset()

Traceback (most recent call last):

File "<string>, line 8, in <module>

fSet.add('v')

AttributeError: 'frozenset' object has no attribute 'add'

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

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.

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

Union( ) is used to combine two sets:

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

numbers = {1,2,3,4}

print(fruits | numbers)

**O/P:**

{1, 2, 3, 4, 'apple', 'cherry', 'banana'}

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

Intersection is used to give common values

Ex:

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

numbers\_and\_strings = {1,2,3,4,"apple"}

print(fruits | numbers\_and\_strings)

**O/P:**

{1, 2, 3, 'apple', 'banana', 4, 'cherry'}

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

Dictionary acts like a map data structure in python. It gives key value pairs. Keys can only be int or string but cannot be list, tuple, set etc whereas values can be anyform.

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

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.

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

Dict = {'Name': 'Geeks', 1: [1, 2, 3, 4]}

print("\nDictionary with the use of Mixed Keys: ")

print(Dict)

**O/P**

Dictionary with the use of Mixed Keys:

{'Name': 'Geeks', 1: [1, 2, 3, 4]}

**Q58. What will the output of the following?**

**var = {}**

**print(type(var))**

<class 'dict'>

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

We can insert values by assigning to it

EX:

thisdict = {

"brand": "Ford",

"model": "Mustang",

"year": 1964

}

thisdict["color"] = "red"

print(thisdict)

**O/P:**

{'brand': 'Ford', 'model': 'Mustang', 'year': 1964, 'color': 'red'}

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

thisdict = {

"brand": "Ford",

"model": "Mustang",

"year": 1964

}

thisdict["color"] = "red"

print(thisdict)

total\_values = thisdict.values()

print("Total values :",total\_values)

**O/P:**

Total values : dict\_values(['Ford', 'Mustang', 1964, 'red'])

**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'])

**O/P:**

John

27

Male

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

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

Syntax: dictionary.get(keyname, value)

Ex:

car = {

"brand": "Ford",

"model": "Mustang",

"year": 1964

}

x = car.get("model")

print(x)

**O/P :** Mustang

**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.

**SYNTAX:** dictionary.items()

EX:

car = {

"brand": "Ford",

"model": "Mustang",

"year": 1964

}

x = car.items()

car["year"] = 2018

print(x)

**O/P:**

dict\_items([('brand', 'Ford'), ('model', 'Mustang'), ('year', 2018)])

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

prime\_numbers = [2, 3, 5, 7]

# remove the element at index 2

removed\_element = prime\_numbers.pop(2)

print('Removed Element:', removed\_element)

print('Updated List:', prime\_numbers)

**O/P:**

Removed Element: 5

Updated List: [2, 3, 7]

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

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.

car = {

"brand": "Ford",

"model": "Mustang",

"year": 1964

}

car.popitem()

print(car)

**O/P:**

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

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

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, see example below.

EX:

car = {

"brand": "Ford",

"model": "Mustang",

"year": 1964

}

x = car.keys()

car["color"] = "white"

print(x)

**O/P:**

dict\_keys(['brand', 'model', 'year', 'color'])

**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.

**eX:**

car = {

"brand": "Ford",

"model": "Mustang",

"year": 1964

}

x = car.values()

car["year"] = 2018

print(x)

**O/P:**

dict\_values(['Ford', 'Mustang', 2018])

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

A loop is an instruction that repeats multiple times as long as some condition is met.

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

Mainly there are two types of loops.

**1) For loop:**

A for loop in Python is used to iterate over a sequence (list, tuple, set, dictionary, and string).

**Syntax:** for iterating\_var in sequence:

statement(s)

**2) While Loop**

The while loop is used to execute a set of statements as long as a condition is true.

**Syntax:** while expression:

statements

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

The for loop is used when we know the number of iterations, that is, how many times a statement must be executed. That is why, when we initialize the for loop, we must define the ending point.

A while loop is used when the number of iterations is unknown. It is used when we need to end the loop on a condition other than the number of repetitions. It is not necessary to know the condition ahead of time in this case. That is why we can use a boolean expression in the loop's initialization.

**In the Absence of Condition**

If no condition is specified in the for and while loop, the loop will iterate infinitely.

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

In Python, break and continue statements can alter the flow of a normal loop.

Loops iterate over a block of code until the test expression is false, but sometimes we wish to terminate the current iteration or even the whole loop without checking test expression.

**CONTINUE:** The continue statement is used to skip the rest of the code inside a loop for the current iteration only. Loop does not terminate but continues on with the next iteration.

**EX:**

for val in "string":

if val == "i":

continue

print(val)

print("The end")

**O/P:**

s

t

r

n

g

The end

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

The break statement terminates the loop containing it. Control of the program flows to the statement immediately after the body of the loop.

If the break statement is inside a nested loop (loop inside another loop), the break statement will terminate the innermost loop

EX:

for val in "string":

if val == "i":

break

print(val)

print("The end")

**O/P:**

s

t

r

The end

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

When the user does not know what code to write, So user simply places pass at that line. Sometimes, pass is used when the user doesn’t want any code to execute. So user can simply place pass where empty code is not allowed, like in loops, function definitions, class definitions, or in if statements. So using pass statement user avoids this error.

def myfunction():

pass

# having an empty function definition like this, would raise an error without the pass statement

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

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.

SYNTAX: range(start, stop, step)

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

**EX:**

statesAndCapitals = {

'Gujarat': 'Gandhinagar',

'Maharashtra': 'Mumbai',

'Rajasthan': 'Jaipur',

'Bihar': 'Patna'

}

print('List Of given states and their capitals:\n')

# Iterating over Values

for state, capital in statesAndCapitals.items():

print(state, ":", capital)

**O/P:**

Gujarat : Gandhinagar

Maharashtra : Mumbai

Rajasthan : Jaipur

Bihar : Patna

**----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------**

**Coding problems**

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

def factorial(n):

if n==1 or n==0:

return 1

result = 1

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

result = result \* i

return result

x = 5

ans = factorial(x)

print("Factorial of",x,"=",ans)

**O/P:**

Factorial of 5 = 120

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

def simple\_interest(P,R,T):

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

return SI

num1 = int(input("enter prinicipal amount"))

num2 = int(input("enter interest time"))

num3 = int(input("enter time period"))

SI\_ans = simple\_interest(num1,num2,num3)

print("The Simple Interest Amount is:",SI\_ans)

**O/P:** enter prinicipal amount -10

enter interest time -20

enter time period -3

The Simple Interest Amount is: 6.0

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

def compound\_interest(P,R,T):

CI = P\*(1+R/100)\*\*T

result1 = CI - P

print("The Compound Interest Amount is:", result1)

num1 = int(input("enter prinicipal amount -"))

num2 = float(input("enter interest rate -"))

num3 = int(input("enter time period -"))

CI\_ans = compound\_interest(num1,num2,num3)

**O/P:**

enter prinicipal amount -1200

enter interest rate -5.4

enter time period -2

The Compound Interest Amount is: 133.0992000000001

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

num = int(input("enter any number:"))

if num > 1:

for i in range(2, num):

if num % i == 0:

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

break

else:

print(num, "- it is a prime number")

else:

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

**O/P:**

enter any number:3

3 - it is a prime number

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

num = int(input("enter any number:"))

temp = num

res = len(str(num))

sum = 0

while num != 0:

ans = num % 10

sum = sum + (ans\*\*res)

num = num//10

if temp == sum:

print(temp,"It is a armstrong number")

else:

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

**O/P:**

enter any number:371

371 It is a armstrong number

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

def fibonacci(n):

a = 0

b = 1

if n < 0:

print("Incorrect input")

elif n == 0:

return a

elif n == 1:

return b

else:

for i in range(2, n):

c = a + b

a = b

b = c

return b

# Driver Program

print(fibonacci(8))

**O/P:**

13

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

def swapping\_list(sl):

sl[0], sl[-1] = sl[-1], sl[0]

return sl

list = [12,3,5,6,9]

print("Before swapping list is:",list)

print("List after swapping is:", swapping\_list(list))

**O/P:**

Before swapping list is: [12, 3, 5, 6, 9]

List after swapping is: [9, 3, 5, 6, 12]

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

temp = 0

def swapping\_list(sl):

temp = sl[0]

sl[0] = sl[1]

sl[1] = temp

return sl

list1 = [8,9,10,'hello']

print("Before swapping list is:",list1)

print("List after swapping is:", swapping\_list(list1))

**O/P:**

Before swapping list is: [8, 9, 10, 'hello']

List after swapping is: [9, 8, 10, 'hello']

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

def myMax(list1):

# Assume first number in list is largest

# initially and assign it to variable "max"

max = list1[0]

# Now traverse through the list and compare

# each number with "max" value. Whichever is

# largest assign that value to "max'.

for x in list1:

if x > max:

max = x

# after complete traversing the list

# return the "max" value

return max

# Driver code

list1 = [10, 20, 4, 45, 99]

print("Largest element is:", myMax(list1))

**O/P:**

Largest element is: 99

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

lst = []

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

# iterating till the range

for i in range(0, n):

ele = int(input("enter elements: "))

lst.append(ele) # adding the element

print(lst)

sum = 0

for i in lst:

sum = sum + i

print("Total sum of elements:", sum)

**O/P:**

Enter number of elements : 5

enter elements: 2

enter elements: 3

enter elements: 4

enter elements: 5

enter elements: 6

[2, 3, 4, 5, 6]

Total sum of elements: 20

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

def palindrome(s):

return s == s[::-1]

num = str(input("enter a string:"))

ans = palindrome(num)

if ans:

print("It is a palindrome")

else:

print("It is not a palindrome")

**O/P**

enter a string:mom

It is a palindrome

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

def strings(s,i):

a = s[0 : i]

b = s[i+1 :]

c = a + b

return c

string = str(input("enter string"))

index = int(input("enter index value of element"))

ans = strings(string, index)

print("String after replacement is:", ans)

**O/P:**

enter stringanusha

enter index value of element2

String after replacement is: ansha

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

def checksubstring(s,s1):

if(s.find(s1)) == -1:

print("No", s1, "is not a substring of",s)

else:

print("Yes", s1, " is a substring of",s)

x = input("enter a string: ")

x1 = input("enter substring: ")

print(checksubstring(x,x1))

**O/P:**

enter a string: lay off

enter substring: off

Yes off is a substring of lay off

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

def string\_k(str1,length):

string = [ ]

new\_string = str1.split(" ")

for index in new\_string:

if len(index) > length:

string.append(index)

return string

str1 = input("enter a string:")

length = int(input("enter length of a string to be checked:"))

print(string\_k(str1,length))

**O/P:**

enter a string:anusha is good

enter length of a string to be checked:2

['anusha', 'good']

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

# initializing dictionary

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

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

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

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

# printing original dictionary

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

# Extract Unique values dictionary values

# Using set comprehension + values() + sorted()

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

# printing result

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

**O/P:**

The original dictionary is : {'gfg': [5, 6, 7, 8], 'is': [10, 11, 7, 5], 'best': [6, 12, 10, 8], 'for': [1, 2, 5]}

The unique values list is : [1, 2, 5, 6, 7, 8, 10, 11, 12]

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

def Merge(dict1, dict2):

res = dict1 | dict2

return res

# Driver code

dict1 = {'x': 10, 'y': 8}

dict2 = {'a': 6, 'b': 4}

dict3 = Merge(dict1, dict2)

print(dict3)

**O/P:**

{'x': 10, 'a': 6, 'b': 4, 'y': 8}

**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\_listoftuples(tup,di):

di = dict(tup)

return di

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

di = { }

print(convert\_listoftuples(tup, di))

**O/P:**

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

**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)]**

# creating a list

list1 = [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)]**

# Creating Tuples and printing values

tuple1 = (7, 2)

tuple2 = (7, 8)

print("First tuple : " + str(tuple1))

print("Second tuple : " + str(tuple2))

# finding all pair Combination of tuples

pairCombi = []

for val1 in tuple1:

for val2 in tuple2:

tup = [val1, val2]

pairCombi.append(tuple(tup))

for val1 in tuple2:

for val2 in tuple1:

tup = [val1, val2]

pairCombi.append(tuple(tup))

# Printing tuple Combination

print("All pair Combinations are : " + str(pairCombi))

**O/P:**

First tuple : (7, 2)

Second tuple : (7, 8)

All pair Combinations are : [(7, 7), (7, 8), (2, 7), (2, 8), (7, 7), (7, 2), (8, 7), (8, 2)]

**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):

# reverse = None (Sorts in Ascending order)

# key is set to sort using second element of

# sublist lambda has been used

return(sorted(tup, key = lambda x: x[1]))

# Driver Code

tup = [('for', 24), ('Geeks', 8), ('Geeks', 30)]

# printing the sorted list of tuples

print(Sort\_Tuple(tup))

**O/P:**

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

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

**\***

**\* \***

**\* \* \***

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

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

for i in range(6):

for j in range(i):

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

print(end="\n")

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

**\***

**\*\***

**\*\*\***

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

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

def triangle(n):

# number of spaces

k = n - 1

# outer loop to handle number of rows

for i in range(0, n):

# inner loop to handle number spaces

# values changing acc. to requirement

for j in range(0, k):

print(end=" ")

# decrementing k after each loop

k = k - 1

# inner loop to handle number of columns

# values changing acc. to outer loop

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

# printing stars

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

# ending line after each row

print("\r")

# Driver Code

n = 5

triangle(n)

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

**\***

**\* \***

**\* \* \***

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

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

def triangle(n):

# number of spaces

k = n - 1

# outer loop to handle number of rows

for i in range(0, n):

# inner loop to handle number spaces

# values changing acc. to requirement

for j in range(0, k):

print(end=" ")

# decrementing k after each loop

k = k - 1

# inner loop to handle number of columns

# values changing acc. to outer loop

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

# printing stars

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

# ending line after each row

print("\r")

# Driver Code

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**

n = 5

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

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

print(j, end = "")

print(end="\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**

rows = int(input("Enter number of rows: "))

ascii\_value = 65

for i in range(rows):

for j in range(i+1):

alphabet = chr(ascii\_value)

print(alphabet, end=" ")

ascii\_value += 1

print("\n")