**# Python Mega Assignment**

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

We call Python as a general-purpose programming language because it is designed to be used in a range of applications, including data science, software and web development, automation, and generally getting stuff done. It’s versatility, along with its beginner-friendliness, has made it one of the most-used programming languages today.

Python is called high-level programming language because it is easy for humans to understand and the coding is also quite easy.

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

In Python, there is no need to define the types of variables. It allows to directly use the variables as its type-checking will be done during the execution of the program. The interpreter checks the program line-by-line and also examines the data type of the variable. So, Python is a dynamically typed language.

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

Some of the main benefits of Python include its ease of use, concise and straightforward syntax, and vast libraries. Other Python advantages are its portability, versatility, large user base, and free & open-source license.

Some of the disadvantages of Python include its slow speed and heavy memory usage. It also lacks support for mobile environments, database access, and multi-threading.

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

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?**

A variable is a named unit of data that is assigned a value. If the value is modified, the name does not change. Python has no command for declaring a variable. A variable is created when some value is assigned to it. The value assigned to a variable determines the data type of that variable. Thus, declaring a variable in Python is very simple.

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

input (): This function first takes the input from the user and converts it into a string. The type of the returned value always will be <class '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 datatype of the value that has been taken as an input using input() function, always will be <class 'str'>.

**Q8. What is type casting?**

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

Example:

b **=** 3.0

print(type(b))

Here, Output will be <class ‘float’>

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

Yes, we can take more than one input from the user with the help of split() function. This function helps in getting multiple inputs from users. It breaks the given input by the specified separator. If a separator is not provided then any white space is a separator. Generally, users use a split() method to split a Python string but one can use it in taking multiple inputs.

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

**Q10. What are keywords?**

Python 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, or variable name. All the keywords in python are written in lowercase except True and False.

Some of the examples of Keywords are: and, or, not, if, if-else, while, break, continue, etc.

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

We cannot use a keyword as a variable name, function name, or any other identifier. They are used to define the syntax and structure of the Python language. If we try to use keywords as a variable name it will show “SyntaxError: invalid syntax” in the output.

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

Python indentation refers to adding white space before a statement to a particular block of code. In another word, all the statements with the same space to the left, belong to the same code block. Indentation is a very important concept of Python because without properly indenting the Python code, you will end up seeing “IndentationError” and the code will not get compiled.

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

With the help of “print()” function, we can throw some output to the screen or any other standard output device.

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

Operators in general are used to perform operations on values and variables. These are standard symbols used for the purpose of logical and arithmetic operations.

Example: +, -, \*, /, <, >, >=, <=, !=, etc.

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

In Python, “/” is a float-division, and “//” is an Integer-division.

when we apply float-division on two numbers, output will be in float. And, when we apply integer-division on two numbers, output will be in integer form. Let’s see an example to understand it better.

a = 5

b = 3

print(“Float-Division on a and b is = ”, a/b)

print(“Integer-Division on a and b is = ”, a//b)

Output:

Float-Division on a and b is = 1.666666666

Integer-Division on a and b is = 1

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

**```**

**iNeuroniNeuroniNeuroniNeuron**

**```**

str1 = "iNeuron" \* 4

print(str1)

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

a = int(input("Please enter your number"))

if a%2 == 0:

    print(a, " is a Even number.")

else:

    print(a, "is a Odd number.")

**Q18. What are Boolean operators?**

Boolean Operators are those that result in the Boolean values of True and False. These include ‘and’, ‘or’ and ‘not’. While ‘and’ and ‘or’ require 2 operands, not is a unary operator. Boolean operators are most commonly used in arithmetic computations and logical comparisons.

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

**1 or 0** = 1

**0 and 0** = 0

**True and False and True** = 0

**1 or 0 or 0** = 1

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

Conditional statements allow you to control the flow of your program based on conditions that you specify. They provide a way to make decisions in your program and execute different code based on those decisions. The following are the conditional statements provided by Python.

* if
* if..else
* Nested if
* if-elif statements.

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

'if', 'elif' and 'else'are conditional statements that provide you with the decision making that is required when you want to execute code based on a particular condition. These statements used in Python helps to automate that decision making process.

**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("Please 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]**

**```**

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

sum = 0

for num in numbers:

    if num%2==0:

        sum = sum + num

    else:

        continue

print("sum of even numbers in the list is =", sum)

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

num1 = int(input("Please enter first number"))

num2 = int(input("Please enter second number"))

num3 = int(input("Please enter third number"))

if num1>num2:

    if num1>num3:

        print(num1, "is the greatest")

    else:

        print(num3, "is the greatest")

elif num2>num3:

    print(num2, "is the greatest")

else:

    print(num3, "is the 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]**

**```**

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

final\_list = []

for num in numbers:

    if num>500:

        break

    elif num>150:

        continue

    elif num%5==0:

        final\_list.append(num)

print(final\_list)

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

In computer programming, a string is a sequence of characters. For example, "hello" is a string containing a sequence of characters 'h', 'e', 'l', 'l', and 'o'. Strings is an immutable data type. Strings in Python can be created using single quotes or double quotes or even triple quotes.

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

In Python, individual characters of a String can be accessed by using the method of Indexing. Indexing allows negative address references to access characters from the back of the String, e.g. -1 refers to the last character, -2 refers to the second last character, and so on. While accessing an index out of the range will cause an IndexError. Only Integers are allowed to be passed as an index, float or other types that will cause a TypeError.

**input:**

string1 = "iNeuron"

print(string1[0], string1[-1])

**output:**

i n

**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"

print(string [-1 : 7 : -1])

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

string = "Big Data iNeuron"

print(string [-1 : : -1])

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

“del” command is used to delete the entire string at once. For example:

string = "Big Data iNeuron"

del string

**Q32. What is escape sequence?**

To insert characters that are illegal in a string, use an escape character. An escape character is a backslash \ followed by the character you want to insert. An example of an illegal character is a double quote inside a string that is surrounded by double quotes

string = 'Welcome to \'iNeuron\’’

print ("By using \: ", string)

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

**```**

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

**```**

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

print(string)

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

Python Lists are just like dynamically sized arrays. In simple language, a list is a collection of things, enclosed in [ ] and separated by commas. List is a sequence data type which is used to store the collection of data. List stores heterogeneous kind of data, makes it powerful tool in Python. A single list may contain Data-Types like Integers, Strings, as well as Objects. Lists are mutable, and hence, they can be altered even after their creation.

Example: list = [‘Big’, ‘Data’, ‘iNeuron’, 2.0, 7]

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

Lists in Python can be created by just placing the sequence inside the square brackets[]. For Example:

#Creating list in python

List1 = ['Big', 'Data', 'iNeuron', 2.0, 7]

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

The index along with the value of the list elements. for example,

list1 = ['Big', 'Data', 'iNeuron', 2.0, 7]

#Accessing the element in a list using for loop.

for num in range (0, len(list1)):

    print(list1[num])

**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.**

list1=[]

quantity = int (input (print ("Enter the quantity of elements you want to enter in the list")))

for num in range (1, quantity+1):

    element = (input(print("please enter", num, "number")))

    list1.append(element)

print("Length of the list is:  ", len(list1))

**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(2, "Big")

print(lst)

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

List and Tuple in Python are the classes of Python Data Structures. The list is dynamic, whereas the tuple has static characteristics. This means that lists can be modified whereas tuples cannot be modified, the tuple is faster than the list because of static in nature. Lists are denoted by the square brackets but tuples are denoted as parenthesis. Lists are mutable but tuples are immutable. The list is better for performing operations, such as insertion and deletion, on the other hand tuple data type is appropriate for accessing the elements

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

To create a tuple we will use () operators.

tuple1 = ("Big", "Data", "Course")

print(tuple1)

print(type(tuple1))

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

tuple1 = ("You're", "welcome", "Mr.")

print(tuple1)

#Trying to add value in tuple using multiple ways

tuple1[3]="Avneesh"

tuple1.append("Avneesh")

tuple1.extend("Avneesh")

tuple1.insert(3, "Avneesh")

print(tuple1)

No, we are unable to add my name in the tuple, because tuples are immutable and,

Tuple does not support item Assignment, and

Tuple object has no attribute append, extend and insert.

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

No, we cannot append two tuple or any other elements in tuple because tuples are immutable and if try to append in tuple it shows error: Tuples object has no attribute append.

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

list1=[]

#Taking tuple values as an input

quantity = int (input (print ("Enter the quantity of elements you want to enter in the tuple")))

for num in range (1, quantity+1):

    element = (input(print("please enter", num, "element")))

    list1.append(element)

#casting list into tuple

tuple2 = tuple(list1)

#Counting and printing the length of input

print("Length of tuple is: ", len(tuple2))

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

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.

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

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

In Python, we create sets by placing all the elements inside curly braces {}, separated by comma. A set can have any number of items and they may be of different types (integer, float, tuple, string etc.). But a set cannot have mutable elements like lists, sets or dictionaries as its elements.

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

#Creating set

set1 = {'Welcome'}

print(set1)

#Adding element

set1.add('iNeuron')

print('After new element addition: ', set1)

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

add() function did not allow to add multiple values in single line, if we try to add them in single line then it will shows an error: “add() takes exactly one argument”.

So, we have to add multiple values one-by-one, using add() function .

set1 = {'Welcome'}

print(set1)

#Adding value one-by-one using add() function

set1.add('Avneesh')

set1.add('Vishal')

set1.add('Shashank')

set1.add('iNeurons')

print(set1)

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

**set.update(\*args)**

It expects a single or multiple iterable sequences as arguments and appends all the elements in these iterable sequences to the set.

**set.add(element)**

It accepts an element as an argument and if that element is not already present in the set, then it adds that to the set.

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

Python Set clear() method removes all elements from the set. The clear() method doesn’t take any parameters.

Syntax: set.clear()

set1 = {22, 'table', 'Data', 156}

print(set1)

set1.clear()

print(set1)

**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.

syntax:

frozenset([elements])

**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.

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

The union of two given sets is the set that contains all the elements of both sets. The union of two given sets A and B is a set that consists of all the elements of A and all the elements of B such that no element is repeated.

set1 = {"cat", 65, "rat", 'dog'}

set2 = {66, 65, 87, 'cat', 'fox'}

print("set1: ", set1)

print("set2: ", set2)

print("Union of set1 and set2 is: ", set1 | set2)

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

The intersection() method returns a new set with elements that are common to all sets.

Example:

set1 = {"cat", 65, "rat", 'dog'}

set2 = {66, 65, 87, 'cat', 'fox'}

print("set1: ", set1)

print("set2: ", set2)

print("Intersection of set1 and set2 is: ", set1 & set2)

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

In Python, dictionaries are mutable data structures that allow you to store key-value pairs. Dictionary can be created using the dict() constructor or curly braces' {}'. Once you have created a dictionary, you can add, remove, or update elements.

dict1 = {

  "Name": "Avneesh",

  "Course": "BigData",

  "Batch": 2.0

}

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

Some of the features that makes dictionary different from all other data structures are,

* A dictionary is a collection of data values.
* It holds a key: value pair in which we can easily access a value if the key is known.
* It improves the readability of your code and makes it easier to debug
* It is fast as the access of a value through a key is a constant time operation

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

dict1 = {

  "Name": "Avneesh",

  "Course": "BigData",

  "Batch": 2.0

}

print("All elements of dictionary: ", dict1)

#Deleting single element in dictionary

del dict1["Batch"]

print("After deleting single element using del: ", dict1)

#Deleting all elements inside dictionary

dict1.clear()

print("After deleting all elements using clear() function: ", dict1)

#Deleting complete dictionary

del dict1

print(dict1)

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

**```**

**var = {}**

**print(type(var))**

**```**

<class dict>

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

dict1 = {

  "Name": "Avneesh",

  "Course": "BigData",

  "Batch": 2.0

}

print(dict1)

#Adding new element in dictionary

dict1["Address"] = 'Ghaziabad'

print('After addition of new element: ', dict1)

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

dict1 = {

  "Name": "Avneesh",

  "Course": "BigData",

  "Batch": 2.0,

  "Address" :'Ghaziabad'

}

#Accessing all the values in dictionary using values() function.

print(dict1.values())

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

dict1 = {

  "Name": "Avneesh",

  "skills": {"python", "sql", "powerBI"}

}

#Accessing the elements of nested dictionary.

print(dict1["skills"])

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

get() function in Python is used to get the value of any specified key from a dictionary. The get() function returns the value of the key if the key is present in the dictionary. The get() function returns a None if the value is not present in the dictionary.

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

In Python Dictionary, items() method is used to return the list with all dictionary keys with values.

dict1 = {

  "Name": "Avneesh",

  "skills": {"python", "sql", "powerBI"}

}

print(dict1.items())

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

List pop in Python is a pre-defined, in-built function that removes an item at the specified index from the list. You can also use pop in Python without mentioning the index value. In such cases, the pop() function will remove the last element of the list.

**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() function in Python is used to retrieve all of the values inside a dictionary. This function doesn't take any parameters and gives a list of values inside a dictionary. If a dictionary has no value, this function returns a null dictionary.

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

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. Example: ‘for loop’, ‘while loop’, ‘nested loop’.

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

Python provides three ways for executing the loops. While all the ways provide similar basic functionality, they differ in their syntax and condition-checking time.

* While loop
* For loop
* Nested loop

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

**For Loop:**

* For loops are designed for iterating over a sequence of items. Ex. list, tuple, etc.
* For loop require a sequence to iterate over.
* For loop is more efficient than a while loop when iterating over sequences, since the number of iterations is predetermined and the loop can be optimized accordingly.

**While Loop:**

* While loop is used when the number of iterations is not known in advance or when we want to repeat a block of code until a certain condition is met.
* While the loop requires an initial condition that is tested at the beginning of the loop.

**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.

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

'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.

**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.

**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 sequences on a sequence of numbers using Python loops.

Example: Printing the numbers from 0 to 4, using range function.

for i in range(5):

    print(i)

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

You can loop through a dictionary by using a for loop.

StatesAndParks = {

    'Kaziranga National Park': 'Assam',

    'Pench National Park': 'Maharastra',

    'Ranthambore National Park': 'Rajasthan',

    'Dhudwa National Park': 'UttarPradesh'

}

print('List Of National Parks:\n')

# Looping over Dictionary

for Parks in StatesAndParks:

    print(Parks)

**### Coding problems**

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

num = int(input(print("Please type a number to find the factorial: ")))

fact = 1

if num == 0 or num == 1:

    print("factorial of ", num, "is :", 1)

else:

    for num in range(1, num+1):

        fact = fact \* num

    print("factorial of ", num, "is :", fact)

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

#Getting Principal, Rate and Time from user.

p = float(input(print("Please enter the principal amount(Rs): ")))

r = float(input(print("Please enter the rate of interest: ")))

t = float(input(print("please enter the time period in years")))

#Calculating Simple Interest using user data.

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

print("Your simple interest on the given data is: ", si)

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

#Getting Principal, Rate and Time from user.

p = float(input(print("Please enter the principal amount(Rs): ")))

r = float(input(print("Please enter the rate of interest: ")))

t = float(input(print("please enter the time period in years: ")))

#Calculating Compound Interest using user data.

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

print("Your compound interest on the given data is: ", ci)

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

#function to find the number is prime or not.

def factorial(num):

    for n in range(2, num):

        if num % n == 0:

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

            break

        else:

            print(num, " is a prime number")

            break

#Getting user input

num = int(input(print("Please enter a number: ")))

if num <= 1:

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

else:

    #Calling function "factorial" with argument.

    factorial(num)

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

#Step3: Typecasting, integer to string and then append elements into list.

def typecasting(num):

    lst = []

    str1 = str(num)

    for i in str1:

        lst.append(i)

    mult2 = Armstrong(lst)

    return mult2

 #Step4: Calculation using for loop, helpful in checking Armstrong number.

def Armstrong(lst):

    mult = 0

    for i in lst:

       mult = mult + int(i)\*\*3

    return mult

#Step1: Getting a user input

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

#step2: Calling function

compare = typecasting(num)

#Final step: After comparison, print the result.

if compare == num:

    print("Yes, ", num, " is a Armstrong Number.")

else:

    print("No, ", num, " is not a Armstrong Number.")

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

# Function to store the Fibonacci number in list and then print.

def Fibonacci(count):

    a = 0

    b = 1

    sum = 0

    lst = []

    for i in range(0, count):

        lst.append(a)

        sum = a + b

        a = b

        b = sum

    print("Fibonacci series: ", lst)

# Getting input from user.

count = int(input(print("Please enter a limit of Fibonacci series: ")))

# Function Call with argument.

Fibonacci(count)

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

#Function to interchange the first and last element of list.

def swap(list1):

    list1.append(list1[0])

    list1[0] = list1[(len(list1)-2)]

    del list1[len(list1)-2]

    return list1

list1 = [65, 'rabbit', 'abacus', 88, 34, 'champion']

print("Original list: ", list1)

#Function Call with argument.

print("After interchange the first and last element: ", swap(list1))

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

#Function to interchange the first and last element of list.

def swap(list1, a, b):

    print("Before Swapping: ", list1)

    list1.append(list1[a-1])

    list1[a-1] = list1[b-1]

    list1[b-1] = list1[len(list1)-1]

    del list1[len(list1)-1]

    return list1

list1 = [65, 'rabbit', 'abacus', 88, 34, 'champion']

print("Original list: ", list1)

a = int(input(print("Please enter first index value to swap.")))

b = int(input(print("Please enter second index value to swap.")))

#Function Call with argument, and printing final list.

print("After Swapping: ", swap(list1, a, b))

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

# finding the largest number in the given list.

def largest(list1):

    a = 0

    for num in list1:

        if num > a:

            lar = num

            a = lar

        else:

            continue

    return a

list1 = [23, 56, 12, 45, 90, 66, 90, 82, 11]

# Calling function, and printing the largest number in the given list.

print("Largest element in the list is: ", largest(list1))

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

# Calculating cumulative sum using 'for' loop, and then return final list to function call.

def cumulativeSum(list1):

    sum = 0

    final = []

    for i in list1:

        sum = sum + i

        final.append(sum)

    return final

list1 = [23, 56, 12, 45, 90, 66, 90, 82, 11]

#Function call with argument list and print final list.

print("Cumulative sum of the given list is: ", cumulativeSum(list1))

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

# Function to reverse the string and checking string is palindrome or not, and finally return the result.

def palindrome(str1):

    reverse = ()

    reverse = str1[::-1]

    if str1==reverse:

        return True

    return False

# Getting input string from user.

str1 = input(print("Please type a string: "))

# Function call with argument string, and then printing result using if-else.

if palindrome(str1):

    print(str1, " is a palindrome string.")

else:

    print(str1, " is not a Palindrome string.")

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

def delIndex(str1, del1):

    finalstr = " "

    for i in range(len(str1)):

        if i != del1-1:

            finalstr = finalstr + str1[i]

        else:

            continue

    return finalstr

str1 = input(print("Please enter a string you want to play with: "))

print("You entered: ", str1)

del1 = int(input(print("Type an index value of the string you want to delete: ")))

print("Your final string after removing nth element is: ", delIndex(str1, del1))

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

# Getting 2 string input from user.

str1 = input(print("Please enter a main string: "))

str2 = input(print("Please enter a substring, to check its presence in previous string: "))

# Checking presence of substring in main string, and return the result.

if str2 in str1:

    print("Yes, substring present in string.")

else:

    print("No, substring not present in string.")

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

# Getting input from user

str1 = input(print("Please enter a sentence: ")).split()

limit = int(input(print("Enter the limit in integer: ")))

# Using for and if-else method to find the result

str2 = ""

for i in range(len(str1)):

    if len(str1[i]) > limit:

        str2 = str2 + str1[i] + ", "

    else:

        continue

# Printing final result.

print("Words found greater than provided length: ", str2)

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

dict1 = {1:'akash', 2:'abhishek', 3:'deepak', 4:'vishal', 5:'vikas'}

dict2 = {1:'vikas', 2:'bunty', 3:'deepak', 4:'sunny', 5:'akash'}

# convert dictionary into set and then assigning dictionary values to sets.

lst3 = set(dict1.values())

lst4 = set(dict2.values())

# Extracting and printing unique values of two dictionaries.

print("Unique values from two dictionaries: ", lst3 & lst4)

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

dict1 = {1:'akash', 2:'abhishek', 3:'deepak', 4:'vishal', 5:'vikas'}

dict2 = {6:'vikas', 7:'bunty', 8:'deepak', 9:'sunny', 10:'akash'}

print("First dictionary: ", dict1)

print("Second dictionary: ", dict2)

(dict2.update(dict1))

print("After merging both dictionaries: ", 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}**

**```**

tup1 = [('Sachin', 10), ('MSD', 7), ('Kohli', 18), ('Rohit', 45)]

dict = {}

for i in range(0, len(tup1)):

    dict[tup1[i][0]] = tup1[i][1]

print("After converting tuple into dictionary: ", dict)

**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)]**

**```**

tup1 = [9, 5, 6]

dict1 = {}

for i in range(0, len(tup1)):

    dict1[tup1[i]] = tup1[i]\*\*3

tup2 = tuple(dict1.items())

print(tup2)

**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)

matrix\_mult1 = [(x,y) for x in test\_tuple1 for y in test\_tuple2]

matrix\_mult2 = [(x,y) for x in test\_tuple2 for y in test\_tuple1]

final = matrix\_mult1 + matrix\_mult2

print(final)

**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)]**

**```**

# Function to shuffle the index value element.

def shuffle(lst1):

    lst1.append(lst1[0])

    lst1[0] = lst1[1]

    lst1[1] = lst1[len(lst1)-1]

    del lst1[len(lst1)-1]

    return (lst1)

tup1 = [('for', 24), ('Geeks', 8), ('Geeks', 30)]

print("Given tuple is: ", tup1)

output = (shuffle(tup1))

print("After shuffle: ", output)

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

**```**

**\***

**\* \***

**\* \* \***

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

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

**```**

for i in range(1, 6):

    print("\* " \* i)

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

**```**

**\***

**\*\***

**\*\*\***

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

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

**```**

n=5

i=1

while(i<=n):

  print(" " \* (n - i) +"\*" \* i)

  i+=1

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

**```**

**\***

**\* \***

**\* \* \***

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

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

**```**

n=5

i=1

while(i<=n):

  print(" " \* (n - i) +"\* " \* i + " ")

  i+=1

**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

i=1

str1 = ""

str2 = ""

while(i<=n):

    str1 = str(i) + " "

    str2+=str1

    print(str2)

    i+=1

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

**```**

**A**

**B B**

**C C C**

**D D D D**

**E E E E E**

**```**

lst1 = ['A', 'B', 'C', 'D', 'E']

j = 1

for i in range(0, 5):

    print((lst1[i] + " ") \* j)

    j+=1