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

Ans. Python is an object-oriented, high-level programming language.Object-oriented means this language is based around objects (such as data) rather than functions, and high-level means it's easy for humans to understand.

Q2. Why is Python called a dynamically typed language?

Ans. Python is both a strongly typed and a dynamically typed language.Strong typing means that variables do have a type and that the type matters when performing operations on a variable.Due to strong typing, types need to be compatible with respect to the operand when performing operations.For example Python allows one to add an integer and a floating point number, but adding an integer to a string produces error.

Dynamic typing means that the type of the variable is determined only during runtime.Due to dynamic typing, in Python the same variable can have a different type at different times during the execution.Dynamic typing allows for flexibility in programming, but with a price in performance.

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

Ans. pros of Python:

Python is free, open-source, and has a vibrant community

Python is easy to learn and read

Python enhances productivity

Python is a portable programming language

Python is an interpreted language

cons of python:

Python is not so strong with mobile computing

Python can have runtime errors

Python consumes a lot of memory space

Python is not easy to test

Q4. In what all domains can we use Python?

Ans. Data Science

Automation

Application Development

AI & Machine Learning

Audio/Video Applications

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

Ans. A variable declaration always contains two components: the type of the variable and its name.

Also, the location of the variable declaration, that is,

where the declaration appears in relation to other code elements,

determines the scope of the variable.

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

Ans. username = input("Enter username:")

print("Username is: " + username)

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

Ans. string

Q8. What is type casting?

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

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

Ans. yes,he split() method is useful for getting multiple inputs from users

#taking three inputs

a,b,c=input("enter three nos:").split()

print("enter first no:",a)

print("enter second no:",b)

print("enter third no:",c)

print()

Q10. What are keywords?

Ans. and,if,in,break,continue,or,elif,else,finally,def

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

Ans. Keywords are predefined, reserved words used in Python programming that have special meanings to the compiler.

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.

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

Ans. 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.

Q13. How can we throw some output in Python?

Ans.a=[1,2,3]

Try:

print(“second element =%d” %(a[1]))

print(“fourth element =%d” %(a[3]))

Except:

print(“an error occurred”)

Q14. What are operators in Python?

Ans.Arithmetic Operators

Comparison Operators

Logical Operators

Bitwise Operators

Identity Operators

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

Ans. / is a floating point division operator and // is an integer division operator.

Float division: gives a decimal answer.

Integer division: gives the answer in whole numbers (the division result is rounded to the nearest whole number).

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

```

iNeuroniNeuroniNeuroniNeuron

```

Ans. multiply\_numeric\_str = "ineuron"\*4

print(multiply\_numeric\_str)

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

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

mod = num % 2

if mod > 0:

print("This is an odd number.")

else:

print("This is an even number.")

Q18. What are boolean operators?

Ans. Boolean is a set of commands that can be used in almost every search engine, database, or online catalogue.

The most popular Boolean commands are AND, OR, and NOT.

Other commands include parentheses, truncation, and phrases.

Q19. What will the output of the following?

```

1 or 0

0 and 0

True and False and True

1 or 0 or 0

```

Ans. print(1 or 0)

print(0 and 0)

print(True and False and True)

print(1 or 0 or 0)

Q20. What are conditional statements in Python?

Ans. Conditional statements are also called decision-making statements.

We use those statements while we want to execute a block of code when the given condition is true or false.

Type of condition statement in Python?

If statement.

If Else statement.

Elif statement.

Nested if statement.

Nested if else statement.

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

Ans. if…elif…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.

The if…elif…else statement used in Python helps 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".

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

if(num>=18):

print("i can vote")

else:

print("i cant vote")

Q23. Write a code that displays the sum of all the even numbers from the given list.

```

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

```

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

print("even nos: ")

for num in numbers:

sum = 0

for num in numbers:

if num%2 == 0:

print(num)

sum = sum + num

print("\nSum of Even Numbers is", sum)

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

Ans.a= int(input("enter first no : "))

b=int(input("enter second no: "))

c=int(input("enter third no: "))

greatest=0

if a>b and a>c:

greatest = a

elif b>c:

greatest = b

else:

greatest = c

print(greatest,"is the greatest of three nos.")

Q25. Write a program to display only those numbers from a list that satisfy the following conditions

- The number must be divisible by five

- If the number is greater than 150, then skip it and move to the next number

- If the number is greater than 500, then stop the loop

```

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

```

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

for i in numbers:

if i > 500:

break

elif i > 150:

continue

elif i % 5 == 0:

print(i)

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

Ans. Strings in python are surrounded by either single quotation marks, or double quotation marks.'hello' is the same as "hello".You can display a string literal with the print() function:

Example:

print("Hello")

print('Hello')

output:Hello

Hello

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

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

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

string = "Big Data iNeuron"

desired\_output = "iNeuron"

Ans. string = "Big Data iNeuron"

print(string)

desired\_output = "iNeuron"

print(desired\_output)

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

string = "Big Data iNeuron"

desired\_output = "norueNi"

Ans. string = "Big Data iNeuron"

print(string)

desired\_output = "norueNi"

print(desired\_output)

Q30. Reverse the string given in the above question.

Ans. string =”Big Data iNeuron”[::-1]

print(string)

Desired\_output = “norueni”[::-1]

print(Desired\_

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

Ans. str1=”prasuna”

Del str1

print(str1)

Q32. What is escape sequence?

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

Q33. How can you print the below string?

'iNeuron's Big Data Course'

Ans. a='iNeuron's Big Data Course'

print(a)

Q34. What is a list in Python?

Ans. Lists are used to store multiple items in a single variable.Lists are one of 4 built-in data types in Python used to store collections of data.

Q35. How can you create a list in Python?

Ans. Lists are created using square brackets:

thislist = ["apple", "banana", "cherry"]

print(thislist)

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

Ans. access the list items by referring to the index number:

Print the second item of the list:

thislist = ["apple", "banana", "cherry"]

print(thislist[1])

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

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

Ans. a=”iNeuron’s Big Data Course’”

print(a)

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

print(list[4][2])

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

Ans. # try block to handle the exception

Try:

my\_list = []

while True:

my\_list.append(int(input()))

# if the input is not-integer, just print the list

except:

print(my\_list)

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

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

Ans. list = ["Welcome", "to", "Data", "course"]

list[3]=”Big”

print(list)

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

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

Q41. How can you create a tuple in Python?

Ans. Tuples are written with round brackets.

thistuple = ("apple", "banana", "cherry")

print(thistuple)

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

Ans.In Python, since tuple is immutable, **you cannot update it**, i.e., you cannot add, change, or remove items (elements) in tuple . tuple represents data that you don't need to update, so you should use list rather than tuple if you need to update it.

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

Ans.Yes,Tuple is immutable, although you can use the + operator to concatenate several tuples. The old object is still present at this point, and a new object is created.

s=(2,5,8)

s\_append = s + (8, 16, 67)

print(s\_append)

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

Ans. #tuple of numbers

numbers=(1,3,4,6,8,9)

#count the no of 1’s in the tuple

count=numbers.count(1)

print(‘the count of 1 is:’,count)

#count the no of 5’s in the tuple

count= numbers.count(5)

print(‘the count of 5 is:’,count)

Q45. What are sets in Python?

Ans. A set is an unordered collection of items. Every set element is unique (no duplicates) and must be immutable (cannot be changed).However, a set itself is mutable. We can add or remove items from it.Sets can also be used to perform mathematical set operations like union, intersection, symmetric difference, etc.

Q46. How can you create a set?

Ans. Sets are written with curly brackets.

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

print(thisset)

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

Ans. set={“apple”,”banana”,”orange”}

set.add(“iNeuron”)

print(set)

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

Ans. a=[1,2,3]

b=[4,5,6]

print(a+b)

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

Ans. Set add():

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. It returns nothing

Set update():

It expects a single or multiple iterable sequences as arguments and appends all the elements in these iterable sequences to the set. It returns nothing

Q50. What is clear() in sets?

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

Q51. What is frozen set?

Ans. Frozen set is just **an immutable version of a Python set object**. While elements of a set can be modified at any time, elements of the frozen set remain the same after creation. Due to this, frozen sets can be used as keys in Dictionary or as elements of another set.

Q52. How is frozen set different from set?

Ans. set:

A set is an unordered and unindexed collection of unique elements. Sets are mutable, you can change the elements using a built-in function like add(), remove(), etc.

Sets can’t be used as a dictionary key or as elements of another set. They can be used as a dictionary value.

Frozen set:

A frozenset is an unordered and unindexed collection of unique elements. It is immutable and it is hashable. It is also called an immutable set. Since the elements are fixed, unlike sets you can't add or remove elements from the set.

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

Ans. The union() method returns a set that contains all items from the original set, and all items from the specified set(s).You can specify as many sets you want, separated by commas.It does not have to be a set, it can be any iterable object.

a={2,3,5}

b={1,3,5}

#compute union of a and b

print(‘a U b=’,a.union (b))

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

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

A={2,3,5}

B={1,3,5}

#intersection btw AandB

print(A.intersection (B))

Q55. What is dictionary in Python?

Ans. Dictionaries are used to store data values in key:value pairs.

A dictionary is a collection which is ordered\*, changeable and do not allow duplicates.

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

Ans. The dictionary Data Structure in Python is **an unordered collection of items**. While other Data Structures use only one value as the element, the dictionary is a slightly more compound data structure. It makes use of two elements i.e. a pair of elements, namely, a key and a value.

Q57. How can we delare a dictionary in Python?

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

Q58. What will the output of the following?

var = {}

print(type(var))

Ans. <class 'dict'>

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

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

CountryCodeDict = {"India": 91, "UK" : 44 , "USA" : 1}

print(CountryCodeDict)

CountryCodeDict["Spain"]= 34

print "After adding"

print(CountryCodeDict)

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

Ans. my\_dict={‘name’:’prasuna’,’age’:’26’}

print(my\_dict)

#output (‘name’:’prasuna’,’age’:’26’)

my\_dict[address]=’hyderabad’

#output(‘address’:’hyderabad’)

print(my\_dict)

@output (‘address’:’hyderabad’,’name’:’prasuna’,’age’:’26’)

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

Ans. In Python, a nested dictionary is a dictionary inside a dictionary. It's a collection of dictionaries into one single dictionary.

people={1:{‘name’:’valli’,’age’:’27’,’gender’:’female’},

2:{‘name’:’prasuna’,’age’:’26’,’gender’:’female’}}

print(people)

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

Ans. The method get() returns a value for the given key. If key is not available then returns default value None.

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

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

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

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

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

Ans. 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.

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

Ans. The Python keys() method is used to retrieve all of the keys from the dictionary. keys() method in Python dictionary returns a list of all the available keys in the dictionary, or an [empty list](https://www.scaler.com/topics/empty-list-in-python/) if the dictionary is empty.

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

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

Q68. What are loops in Python?

Ans. for loop,while loop,nested loop

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

Ans. three types of loops in python.they are

1) For loop

2) While loop

3) Nested loops

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

Ans. for loop execute n no of steps perform in single line

for(initialization;condition;inc/dec)

{

stmnts;

}

While loop execute n no of steps in more than one line.

while(condition)

{

Stents;

inc/dce;

}

Q71. What is the use of continue statement?

Ans. 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?

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

Q73. What is the use of pass statement?

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

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

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

Q75. How can you loop over a dictionary?

Ans. You can loop through a dictionary **by using a for loop**. When looping through a dictionary, the return value are the keys of the dictionary, but there are methods to return the values as well.

### **Coding problems**

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

Ans. num=int(input(“Enter a number: ”)

Factorial =1

If num <0:

print(“factorial does no exit negative numbers”)

Elif num==0:

print(“the factorial of 0 is 1”)

Else

For i in range(1,num+1):

factorial=factorial\*i:

print(“the factorial of”,num,”is”,factorial)

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

Ans. print(“Enter the principle amount: ” )

p=int(input())

print(“enter the rate of interest(%): “)

r=float(input())

print(“enter time period: “)

t=float(input))

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

print(\n”simple interest amount: “)

print(si)

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

Ans. print(“Enter the principle amount: “)

p=float(input())

print(“enter the rate of interest: “)

r+float(input())

print(“enter the time period: “)

ci=P\*(pow((1+r/100),t))

print(“\n compound interest: “)

print(ci)

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

Ans. num=int(input(“Enter the number: “)

If num >1:

For i in range(2,int(num/2)+1):

If (num%i==0):

print(num,”is not a prime no: “)

Break

Else:

print(num,”is a prime no: “)

Q80. Write a Python program to check Armstrong Number.

Ans. num= (int(input(“enter the number”)

sum=0

temp=num

While temp>0:

digit=temp%10

sum+=digit\*\*3

temp//=10

If num==sum

print(num,”is an armstrong no”)

Else:

print(num,”is not an armstrong no”)

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

Ans. def Fibonacci(n):

if n<0:

print("Fibonacci can't be computed")

# First Fibonacci number

elif n==1:

return 0

# Second Fibonacci number

elif n==2:

return 1

else:

return Fibonacci(n-1)+Fibonacci(n-2)

n=10

print(Fibonacci(n))

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

Ans. def swapList(newList):

size = len(newList)

# Swapping

temp = newList[0]

newList[0] = newList[size - 1]

newList[size - 1] = temp

return newList

# Driver code

newList = [12, 35, 9, 56, 24]

print(swapList(newList))

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

Ans. def swapPositions(list, pos1, pos2):

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

return list

# Driver function

List = [23, 65, 19, 90]

pos1, pos2 = 1, 3

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

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

Ans. def Nmax Elements(list1, N):

final\_list = []

for i in range(0, N):

max1 = 0

for j in range(len(list1)):

if list1[j] > max1:

max1 = list1[j];

list1.remove(max1);

final\_list.append(max1)

print(final\_list)

# Driver code

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

N = 2

# Calling the function

Nmax Elements(list1, N)

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

Ans. def Cumulative(lists):

cu\_list = []

length = len(lists)

cu\_list = [sum(lists[0:x:1]) for x in range(0, length+1)]

return cu\_list[1:]

# Driver Code

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

print (Cumulative(lists))

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

Ans. def isPalindrome(s):

return s == s[::-1]

# Driver code

s = "malayalam"

ans = isPalindrome(s)

if ans:

print("Yes")

else:

print("No")

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

Ans. test\_str = "prasunakondapalli"

# Removing char at pos 3

new\_str = ""

for i in range(len(test\_str)):

if i != 2:

new\_str = new\_str + test\_str[i]

# Printing string after removal

print ("The string after removal of i'th character : " + new\_str)

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

Ans. string="StudyTonight"

substring="Study"

print("Does ",substring," exist in ",string,"?")

if substring in string:

print("Yes")

else:

print("False")

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

Ans. def word\_k(k, s):

# split the string where space comes

word = s.split(" ")

# iterate the loop for every word

for x in word:

# if length of current word

if len(x)>k:

# greater than k then

print(x)

k = 3

s ="Python is good"

word\_k(k, s)

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

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

'B' : [4, 6, 8, 10],

'C' : [6, 12, 4 ,8],

'D' : [5, 7, 2]}

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

# Using list comprehension, values() and sorted()

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

# print result

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

Q91. Write a Python program to merge two dictionary.

Ans. dict\_1 = {1: 'a', 2: 'b'}

dict\_2 = {2: 'c', 4: 'd'}

print(dict\_1 | dict\_2)

Q92. Write a Python program to convert a list of tuples into dictionary.

Input : [('Sachin', 10), ('MSD', 7), ('Kohli', 18), ('Rohit', 45)]

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

Ans. # initializing the list

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

# converting to dict

result = dict(tuples)

# printing the result

print(result)

Q93. Write a Python program to create a list of tuples from given list having number and its cube in each tuple.

Input: list = [9, 5, 6]

Output: [(9, 729), (5, 125), (6, 216)]

Ans. my\_list = [9, 5, 6]

print("The list is ")

print(my\_list)

my\_result = [(val, pow(val, 3)) for val in my\_list]

print("The result is ")

print(my\_result)

Q94. Write a Python program to get all combinations of 2 tuples.

Input : test\_tuple1 = (7, 2), test\_tuple2 = (7, 8)

Output : [(7, 7), (7, 8), (2, 7), (2, 8), (7, 7), (7, 2), (8, 7), (8, 2)]

Ans. from itertools import chain, product

# tuples

test\_tuple1 = (7, 2)

test\_tuple2 = (7, 8)

# original tuples

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

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

# All pair combinations of 2 tuples

result = list(chain(product(test\_tuple1, test\_tuple2), product(test\_tuple2, test\_tuple1)))

# result

print("The resultant tuple : " + str(result))

Q95. Write a Python program to sort a list of tuples by second item.

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

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

Ans. #tuple list

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

print("Orignal Tuple List :" ,tuple)

#function

def Sort(tuple):

# Sorts in Ascending order

tuple.sort(key = lambda a: a[1])

return tuple

# printing the sorted list of tuples

print("Sorted Tuple List:" ,Sort(tuple))

Q96. Write a python program to print below pattern.

\*

\* \*

\* \* \*

\* \* \* \*

\* \* \* \* \*

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

for i in range(rows):

for j in range(i+1):

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

print("\n")

Q97. Write a python program to print below pattern.

\*

\*\*

\*\*\*

\*\*\*\*

\*\*\*\*\*

Ans. # Python 3.x code to demonstrate star pattern

# Function to demonstrate printing pattern

def pypart2(n):

# number of spaces

k = 2\*n - 2

# 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 - 2

# 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")

n = 5

pypart2(n)

Q98. Write a python program to print below pattern.

\*

\* \*

\* \* \*

\* \* \* \*

\* \* \* \* \*

Ans. # Function to demonstrate printing pattern triangle

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

for j in range(0, k):

print(end=" ")

# decrementing k after each loop

k = k - 1

# inner loop to handle number of columns

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

# printing stars

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

print("\r")

n = 5

triangle(n)

Q99. Write a python program to print below pattern.

1

1 2

1 2 3

1 2 3 4

1 2 3 4 5

Ans. # Function to demonstrate printing pattern of numbers

def numpat(n):

num = 1

# outer loop to handle number of rows

for i in range(0, n):

num = 1

# inner loop to handle number of columns

# values changing acc. to outer loop

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

# printing number

print(num, end=" ")

# incrementing number at each column

num = num + 1

# ending line after each row

print("\r")

n = 5

numpat(n)

Q100. Write a python program to print below pattern.

A

B B

C C C

D D D D

E E E E E

Ans.# Function to demonstrate printing pattern of alphabets

def alphapat(n):

# initializing value corresponding to 'A'

# ASCII value

num = 65

for i in range(0, n):

# inner loop to handle number of columns

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

ch = chr(num)

print(ch, end=" ")

num = num + 1

print("\r")

n = 5

alphapat(n)