**Q1. Python as a general purpose and high-level programming language --**

Python is called as high-level programming language because it is “easy to understand by humans and also it is in a human readable format”.

**Q2. Python called a dynamically typed language**

Suppose if we don’t declare the type of variable in python it doesn’t have any problem. It states the type during the run time of the program, and it takes the responsibility of memory management. So, it is called as dynamically typed language.

**Q3. Pros & Cons of Python**

|  |  |
| --- | --- |
| **Pros** | **Cons** |
| * Free & Open-source license * Easy to use, understand, and beginner friendly. * Straight-forward syntax. * Programmer doesn’t need to worry about declaring the variables and their data types * **Larger libraries support** - so we can find almost all the functions we need and we don’t want to depend on other external libraries. | * Slow execution speed. * Uses the large amount of memory. * Slow processing power so python is weak in mobile computing. * Since python is a dynamically typed language it occurs run-time errors. * Testing in python takes more time compared with other languages. |

**Q4.Domains we use python**

Python is used in multiple domains such as **Data scientist, Machine learning, deep learning, networking, artificial intelligence…**

**Q5. Declaring variables**

Variable is a name given to a memory location and acts as a container to store values

Declaring variable: first\_name = ‘Aswanth’

**Q6.Taking input from user**

first\_name = input (“Enter your first name: “)

second\_name = input (“Enter your second name: “)

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

**STRING Data type**

**Q8.Type casting -** The conversion of one data type into another data type is known as type casting.

The type casting is of 2 types

**Implicit** – In this, python converts data type into another data type automatically.

#Implicit type casting

# automatically converts a to int

a = 4

print(type(a)) # int

# automatically converts

# b to float

b = 2.5

print(type(b))  #<class 'float'>

# automatically converts c to float as it is float addition

c = a + b

print("c =",c)

print(type(c))  #<class 'float'>

**Explicit –** In this , the user converts the data type into another data type.

#Explicit type casting

num\_int = 10

num\_str = '25'

print(type(num\_int))    #<class 'int'>

print(type(num\_str))    #<class 'str'>

casted\_num\_str = int(num\_str)

total\_sum = num\_int + casted\_num\_str

print("total sum = ",total\_sum) #total sum =  35

print(type(total\_sum))  #<class 'int'>

#example-2

a = 3.6

#typecast of float to int

b =  int(a)

print("b = ",b) #b =  3

print("type of b : ",b) #type of b :  3

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

Yes, it is possible to take more than one input from the user. It is possible by using the split() function method

#using split() fncn

x,y,z = input("Enter three values \_\_ \_\_ \_\_ ").split()   #Enter three values \_\_ \_\_ \_\_ VIRAT KOHLI CRICKETER

print("Enter your first name :",x)  #Enter your first name : VIRAT

print("Enter your second name :",y) #Enter your second name : KOHLI

print("Enter your Profession:",z)   #Enter your Profession: CRICKETER

#OUTPUT :

# Enter three values \_\_ \_\_ \_\_ VIRAT KOHLI CRICKETER

# Enter your first name : VIRAT

# Enter your second name : KOHLI

# Enter your Profession: CRICKETER

**Q10. keywords**

Keywords are pre-defined reserved words that have a special meaning to the compiler. We cannot use a keyword as a variable name or function names. Some of the keywords are (True, False, and, Break, Continue)

**Q11**. **Can we use keywords as a variable?**

Keywords cannot be used as a variable name because they are used to define the syntax of coding.

and = 6

print(and) #SyntaxError: invalid syntax

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

Indentation in Python refers to adding white spaces before the statement to a particular block of code and it is used to indicate the block of code.

**Q13.Throwing output in python-** We throw output by using the **print( )** function.

**Q14.Operators in python:**

Python operators are used to perform the operations on variables and values. Operators are divided into,

Arithmetic - used with numeric values to perform common mathematical operations. (+,-,\*,%,/,//)

Assignment – assigning the values to variables (x=5, y += 10 => y = y+10)

Comparison – compares the two values and returns Boolean value(x = = y, a < b, c > d, y != z)

Logical – used to check whether the condition is True or False (and, or, not )

#Arithmetic Operators

a = 10

b = 20

print("Addition of a and b : ",a+b)

print("Subtraction of a and b : ",a-b)

print("Multiplication of a and b : ",a\*b)

print("Division of a and b : ",a/b)

print("Modulo of a and b : ",a%b)

print("Floor Division of a and b : ",a//b)

#Assignment operator

a = 5

b = a + 5 # 5+5 =10

b+=a # b=b+a => 10+5 = 15

print("value of a and  b : ",a,b)

#comparison operator

a = 10

b = 3

print(a>b) # True

#Logical operator

x = 50

y = 80

print((x>y)and(y<x)) #False

print((x<y)or(x>y)) #True

print(not(x>y)) #True

**Q15.** **difference between / and // operators**

/ - Division operator which performs division.

# / **Division Operation**

a= 47

b = 4

print("Division of a and b : ",a/b)  #Division of a and b :  11.75

**#Floor division**

print("Floor Division of a and b : ",a//b)  #Floor Division of a and b :  11

// - floor division gives the quotient value as rounded , if the quotient is positive (eg.2.57)it gives the value as 2, suppose it is negative (eg,-2.56) it gives as -3.

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

**```iNeuroniNeuroniNeuroniNeuron**

str\_1 = 'iNeuron'

str\_2 = 4 \* str\_1

print(str\_2)

Q**17. Write a code to take a number as an input from the user and check if the number is odd or even.**

#get no. from user

num\_1 = int(input("Entera a number : "))

#check no. is odd or even

if (num\_1%2 == 0):

    print("The number is EVEN")

else:

    print("The number is ODD")

**Q18. What are Boolean operator?**

Boolean operator used to check the conditions whether True or False by using and, or, not

b = 10

c = 50

print((b>c)and(c<b)) #False

print((b<c)or(b>c)) #True

print(not(b>c)) #True

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

```

**1 or 0 ---🡪 1**

0 and 0 **---🡪 0**

True and False and True **---🡪 False**

1 or 0 or 0 **---🡪 1**

```

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

Conditional statements are defined as it tells the computer to execute certain block of code when it is true suppose if the condition is false it executes another block of code.

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

These are the conditional statements used when you want to execute the block of code based on the certain conditions.

**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".**

# take input age of person from user

age = int(input("Enter the 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]

result = 0

for num in numbers:

  if num%2 == 0:

    result = result+num

  else:

    continue

print("sum of even numbers : ",result)

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

a = int(input("Enter the value of a :"))

b = int(input("Enter the value of b :"))

c = int(input("Enter the value of c :"))

if (a>b) and (a>c):

  print("a is the greatest number")

elif (b>c) and (b>a):

  print("b is the greatest number")

else:

  print("c is the greatest number")

**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]

i = []

for num in numbers:

  if num > 150:

    if num > 500:

      break

  elif num%5==0:

    i.append(num)

print(i)

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

Strings are the characters that comes within the quotations(“ “ or ‘ ‘) single or double quotations.

my\_var = "Hey Buddy!!"

print(type(my\_var))

# output

<class 'str'>

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

We can access the string by using the brackets [ ].It can be accessed by the index

a = 'hello world'

print(a[2])

o/p => l

**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:])

# desired\_output = "iNeuron"

**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:-8:-1])

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

string = "Big Data iNeuron"

print(string[::-1])

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

By using “**del”** keyword, we can delete the entire string.

str\_1= "Cricketer"

str\_2 = "Dhoni"

del(str\_1)

**Q32. What is escape sequence?**

An escape sequence is a sequence with the special meaning when it is used inside the strings. A “backslash” (\) sequence acts as a escape character inside a string.

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

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

str\_1 =  " \'iNeuron's Big Data Course' "

print(str\_1)

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

List is a collection of multiple data’s enclosed in a square brackets [ ] in a single variable and accessed by using indexing, first element as [0] and second element as [1] which are separated by commas.

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

Lists are created by using the [ ] square brackets.

mylist =["Dhoni","Shewag","Yuvaraj","Virat"]

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

We can access the list by using indexing,

mylist =["Dhoni","Shewag","Yuvaraj","Virat"]

print(mylist[2])

# output : Yuvaraj

**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.**

mylst = input("Enter the number of elements separated by the space: ").split(" ")

print("The length of the string is : ",len(mylst))

**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?**

Since like a list, tuple is also used to store multiple values in a ( ) brackets. But the tuple is **ordered** and **unchangeable**. Once the tuple is created it cannot be changed.

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

Tuple is created by using the () parentheses.

mytupl = (“orange”,667,898,456)

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

**No,** I’m not able to add my name because tuple does not accept the attribute append and tuples are **immutable.** It means that we cannot create, add, or remove any items once the tuple is created. But we can do it by typecast method.

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

mytuple.append("aswanth")

print("mytuple")

# AttributeError: 'tuple' object has no attribute 'append'

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

**Yes,** two tuples can be appended by using concatenation.

tup1 = ("Aswanth")

tup2 = ("Krishna")

mytup = tup1 + tup2

print(mytup)

# output : AswanthKrishna

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

tup1 = input("Enter values separeted by spaces: ").split(" ")

tup1 = tuple(tup1)

print("The length of the tuple is :",len(tup1))

#Output :

# Enter values separeted by spaces: raj krisna aswanth abi

# The length of the tuple is : 4

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

Sets are the collection of multiple values which are **unordered, unchangeable** andhas no **duplicate** values.

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

Set is created with the curly braces { }.

set1 = {"python","Java","C#"}

print(set1)

# output: {'C#', 'python', 'Java'}

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

set1 = {"iNeuron"}

print(set1)

#Output : {'iNeuron'}

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

set1 = set()

set1.add("Dhoni")

set1.add("Dravid")

set1.add("Sachin")

set1.add("Virat")

set1.add("Yuvi")

print(set1)

#Output : {'Sachin', 'Dhoni', 'Virat', 'Yuvi', 'Dravid'}

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

**add( )**  function is used to add a single element whereas by using **update( )** function we can add multiple elements.

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

**Clear( )**  is used to remove all the elements in the set.

**Q51. What is frozen set?**

Frozen set ( ) is a function that returns the unchangeable frozen set object like set object which is unchangeable.it is freeze and unchangeable. It is also used as a key in the dictionary.

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

|  |  |
| --- | --- |
| **Frozen set** | **Set** |
| * Immutable | * Mutable |
| * Used as keys | * cannot be used as keys |

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

**union()**  set is a method that returns or combines all the values containing in both the sets.

set1 = {1,2,3,4}

set2 = {3,5,6,7,8}

myset = set1 | set2

print(myset)

# output : {1, 2, 3, 4, 5, 6, 7, 8}

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

**intersection( )** is the method that returns the common values from the sets.

set1 = {1,2,3,4}

set2 = {2,3,5,6,7,8}

myset = set1.intersection(set2)

print(myset)

#  output : {2, 3}

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

Dictionary is a collection which is ordered, changeable and won’t allow duplicates. It is stored as key and values and referred by the key name.

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

Dictionary contains both the **key** and **values** whereas the other data structures have only a single value, no **keys.**

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

Dictionary is declared by the curly brackets { }.

dict1 = { }

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

**var = {}**

**print(type(var)) - Dict**

**Ans : < class 'dict' >**

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

dict1 = {

    'car':"Scorpio",

    'Company': "Mahindra",

    'Year' : "2022"

}

dict1["color"] = "Red"

print(dict1)

#output :{'car': 'Scorpio', 'Company': 'Mahindra', 'Year': '2022', 'color': 'Red'}

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

dict1 = {

    'car':"Scorpio",

    'Company': "Mahindra",

    'Year' : "2022",

    'color' : "Red"

}

for x,y in dict1.items():

    print(f"key : {x} and value : {y}")

    # Output:

    # key : car and value : Scorpio

    # key : Company and value : Mahindra

    # key : Year and value : 2022

    # key : color and value : Red

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

dict2 = {

    'car name':"Fortuner",

    'color':{'model-abc':"Red",'mode-xyz': "Whilte"},

    'Manufacturer': "Toyota"

}

for x,y in dict2['color'].items():

    print(f"key : {x} and value : {y}")

# Output:

key : model-abc and value : Red

key : mode-xyz and value : Whilte

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

**get ( )** it returns the value with the specified key

dict2 = {

    'car name':"Fortuner",

    'color':{'model-abc':"Red",'mode-xyz': "Whilte"},

    'Manufacturer': "Toyota"

}

print(dict2.get('Manufacturer')) # Toyota

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

**items( )** returns the key- value pairsin the dictionary .

dict1 = {

    'car':"Scorpio",

    'Company': "Mahindra",

    'Year' : "2022",

    'color' : "Red"

}

for x,y in dict1.items():

    print(f"key : {x} and value : {y}")

    # Output:

    # key : car and value : Scorpio

    # key : Company and value : Mahindra

    # key : Year and value : 2022

    # key : color and value : Red

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

**pop()** removes the specified value from the dictionary.

dict1 = {

    'car':"Scorpio",

    'Company': "Mahindra",

    'Year' : "2022",

    'color' : "Red"

}

dict1.pop('Company')

print(dict1)

# Output:{'car': 'Scorpio', 'Year': '2022', 'color': 'Red'}

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

**popitems()** removes the item that is lastly inserted in the dictionary and returns the item which is removes from the dictionary.

dict2 = {

    'car name':"Fortuner",

    'color':{'model-abc':"Red",'mode-xyz': "Whilte"},

    'Manufacturer': "Toyota"

}

a = dict2.popitem()

print(a)    #('Manufacturer', 'Toyota')

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

keys() returns the view object and contains the keys of dictionary as a list.

dict3 = {

  "brand": "Hyundai",

  "model": "i20",

  "year": "2020"

}

x = dict3.keys()

print(x)  #dict\_keys(['brand', 'model', 'year'])

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

values() returns the view object and contains the values of dictionary as a list.

dict3 = {

  "brand": "Hyundai",

  "model": "i20",

  "year": "2020"

}

x = dict3.values()

print(x)  #dict\_values(['Hyundai', 'i20', '2020'])

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

Loops are used to repeat the certain block of code for many times and in python it is of 2 types **for** and **while loops.**

**Q69. How many types of loops are there in Python?**

* **for** loop
* **while** loop

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

**for -** used when the number of iterations is **known**.

**While** – used when number of iterations are **unknown**

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

used to end the current iteration in a loop and it stops the execution of loop from after the continue statement and forces to the next iteration.

for var in "Big Data":

    if var == "a":

        continue

    print(var)

# output

B

i

g

D

t

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

**break statement** is used to terminate the execution of the loop

for i in range(1,20):

    print(i)

    if i ==7:

        break

# output

1

2

3

4

5

6

7

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

Used as a placeholder for the future code. If an empty function or loop is declared an error occurs but if we use pass it neglects the error.

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

**Range function** gives the sequence of numbers starting from 0 by default and increases by 1 and ends the function before the specified number.

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

dict1 = {

    'car':"Scorpio",

    'Company': "Mahindra",

    'Year' : "2022",

    'color' : "Red"

}

for i in dict1:

    print(i)

#output

car

Company

Year

color

**CODING PROBLEMS**

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

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

factorial = 1

if num < 0 :

    print("Enter the positive integer :")

elif num == 0:

    print("The Factorial of 0 is 1 ")

else:

    for i in range(1,num+1):

        factorial = factorial\*i

    print("The fatorial of",num,"is",factorial)

# output

# Enter a number : 4

# The fatorial of 4 is 24

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

def simple\_interest(p,t,r):

    print("Enter the principal amount :",p)

    print("Enter the time period :",t)

    print("Enter the rate of interest: ",r)

    si =(p \* t \* r)/100

    print("The simple interest is ",si)

    return si

simple\_interest(1000,4,7)

#Output

Enter the principal amount : 1000

Enter the time period : 4

Enter the rate of interest:  7

The simple interest is  280.0

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

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

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

    ci = amount - principal

    print("Compound interest is ",ci)

#Input from user

principal = int(input("Enter the prinipal amount :"))

rate = float(input("enter the rate:"))

time = float(input("Enter the time period :"))

compound\_interest(principal,rate,time)

# O/P

Enter the prinipal amount :2000

enter the rate:2.5

Enter the time period :4.6

Compound interest is  240.5764992163422

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

from math import sqrt

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

prime\_flag = 0

if(num > 1):

    for i in range(2, int(sqrt(num)) + 1):

        if (num % i == 0):

            prime\_flag = 1

            break

    if (prime\_flag == 0):

        print("True",num," is a prime number")

    else:

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

else:

    print("False")

#O/P

Enter the number :97

True 97  is a prime number

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

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

s = num

l = len(str(num))

sum1 = 0

while num != 0:

    r = num % 10

    sum1 = sum1+(r\*\*l)

    num = num//10

if s == sum1:

    print("The given number", s, "is armstrong number")

else:

    print("The given number", s, "is not armstrong number")

#O/P

Enter the number :345

The given number 345 is not armstrong number

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

def Fibonacci(n):

    if n<= 0:

        print("Incorrect input")

    elif n == 1:

        return 0

    elif n == 2:

        return 1

    else:

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

print(Fibonacci(12)) # o/p: 89

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

def swaplist(newlist):

    newlist[0],newlist[-1] = newlist[-1],newlist[0]

    return newlist

newlist = [100, 67, 34, 76, 99]

print(swaplist(newlist))

# O/P => [99, 67, 34, 76, 100]

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

def N\_max\_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)

list1 = [85, 10, 34, 2, 100, 3, 67, 23, 6]

N = 3

N\_max\_elements(list1, N)

#O/P : [100, 85, 67]

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

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

new\_list=[]

j=0

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

    j+=list[i]

    new\_list.append(j)

print(new\_list)

#O/P : [10, 30, 60, 100, 150]

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

def isPalindrome(m):

    return m == m[::-1]

m = "rotator"

ans = isPalindrome(m)

if ans:

    print("Yes the given string is palindrome")

else:

    print("No the given string is not a palindrome")

 # O/P => Yes the given string is palindrome

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

def remove(string, i):

    a = string[: i]

    b = string[i + 1:]

    return a + b

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

    string = "BIGDATAANALYTICS"

    i = 9

    print(remove(string, i))

    #O/P => BIGDATAANLYTICS

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

string1 = "Big Data Analytics"

if "Data" in string1:

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

else:

    print("No! it is not present")

#O/p : Yes! it is present in the string

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

def string\_k(k, str):

    string = []

    text = str.split(" ")

    for x in text:

        if len(x) > k:

            string.append(x)

    return string

k = 5

str = "For every action there is an equal and opposite reaction"

print(string\_k(k, str))

 #o/p = > ['action', 'opposite', 'reaction']

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

dict\_1 = {'always' : [9, 8, 7, 5],

            'be' : [1, 3, 8, 15],

            'happy' : [11, 12, 14, 10],

            'nanba' : [20, 4, 0]}

print("The original dictionary is : " + str(dict\_1))

# Extract Unique values dictionary values

x=[]

for i in dict\_1.keys():

    x.extend(dict\_1[i])

x=list(set(x))

x.sort()

# printing result

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

#O/P

The original dictionary is : {'always': [9, 8, 7, 5], 'be': [1, 3, 8, 15], 'happy': [11, 12, 14, 10], 'nanba': [20, 4, 0]}

The unique values list is : [0, 1, 3, 4, 5, 7, 8, 9, 10, 11, 12, 14, 15, 20]

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

#Merge dictionary

dict\_1 = {

    'always' : [9, 8, 7, 5],

    'be' : [1, 3, 8, 15]

}

dict\_2 = {

    'happy' : [11, 12, 14, 10],

    'nanba' : [20, 4, 0]

}

res = str(dict\_1) + str(dict\_2)

print("The merged dictionary is :",res)

#O/p :

The merged dictionary is : {'always': [9, 8, 7, 5], 'be': [1, 3, 8, 15]}{'happy': [11, 12, 14, 10], 'nanba': [20, 4, 0]}

**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}**

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

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

print (dict([('Sachin', 10), ('MSD', 7), ('Kohli', 18), ('Rohit', 45)]))

# 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)]**

lst = [9,5,6]

res = [(val, val\*\*3) for val in lst]

print("Number and its cube :",res)

#o/p :

Number and its cube : [(9, 729), (5, 125), (6, 216)]

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

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

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

test\_tuple1 = (7,2)

test\_tuple2 = (7,8)

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

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

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

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

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

#O/P

The original tuple 1 : (7, 2)

The original tuple 2 : (7, 8)

The filtered tuple : [(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):

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

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

print(Sort\_Tuple(tup))

#o/p

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

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

**\***

**\* \***

**\* \* \***

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

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

def star(n):

    for i in range(0, n):

        for j in range(0, i+1):

            print("\* ",end="")

        print("\r")

n = 5

star(n)

#o/p

\*

\* \*

\* \* \*

\* \* \* \*

\* \* \* \* \*

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

**\***

**\*\***

**\*\*\***

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

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

n = 5

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

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

    #O/P

    \*

   \*\*

  \*\*\*

 \*\*\*\*

\*\*\*\*\*

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

**\***

**\* \***

**\* \* \***

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

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

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

k = 0

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

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

        print(end="  ")

    while k!=(2\*i-1):

        print("\* ", end="")

        k += 1

    k = 0

    print()

O/P:

\*

\* \* \*

\* \* \* \* \*

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

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

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

**1**

**1 2**

**1 2 3**

**1 2 3 4**

**1 2 3 4 5**

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

for i in range(rows):

for j in range(i+1):

print(j+1, end=" ")

print("\n")

#O/P:

1

1 2

1 2 3

1 2 3 4

1 2 3 4 5

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

**A**

**B B**

**C C C**

**D D D D**

**E E E E E**

def alphapat(n):

    num = 65

    for i in range(0, n):

        for j in range(0, i+1):

            ch = chr(num)

            print(ch, end=" ")

        num = num + 1

        print("\r")

n=5

alphapat(n)

#O/P :

A

B B

C C C

D D D D

E E E E E