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

➔ Python is a general purpose and HLL because it is a human readable and can be used for building software in a wide variety of application domains.

Q2. Why is Python called a dynamically typed language?

➔ Because the type of data is defined during runtime, we don't need to assign a data type to a variable during declaration.

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

➔ Pros are:
Dynamically typed.
Easy to understand.
Large varieties of libraries and frameworks.

➔ Cons are:
Slow processing.
Majorly it is not used for mobile development.
Major chance of runtime error.

Q4. In what all domains can we use Python?

➔ Artificial Intelligence, Machine Learning, Data Science, Web development, Game Development etc.

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

➔ Variable are named memory location used to point object. We can declare them using character, Alpha numeric and underscore (a-z, A-Z, 0-9 or with _).

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

➔ Var = input() or var = int(input()) we can use different data type also.

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

➔ String.

Q8. What is type casting?

➔ Type casting means converting one data type to another data type.

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 multiple input using single input() function by using split() method.
x,y,z = input().split()

Q10. What are keywords?

➔ Keywords are reserved word that are predefined for specific purpose and cannot be used for variable, function and identifier declaration.
Example – if, else, elif, int, for, continue etc.

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

➔ No, we cannot use keyword as a variable.

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

➔ Indentation is used in different programming language for better readability.
In Python, Indentation represents a certain block of code. We give Indentation in Python using a tab or 4 space.

Q13. How can we throw some output in Python?

➔ By using print() function.

Q14. What are operators in Python?

➔ Operators are used to perform different computational operation. Example -> +, -, /, //, ==.

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

➔ / is a standard division it will always give a float value.
// is a floor division it will always give an integer value and will round off the value to next smallest integer.

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

➔ print(4*"iNeuron")

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

➔
```
num = int(input("Enter a Number = "))      #num is variable
if num % 2 == 0:
   print("Even Number")
else:
   print("Odd Number")
```

Q18. What are boolean operator?

➔ Boolean operator is built in data type. Boolean operators are True and False.

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 statement in Python is used to perform conditional operation based on True or False.

```
if
 statement
else
 statement
```

Nested if else

```
if
 statement
elif
 statement
else
 statement
```

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

➔ **if** -> is used to check for true condition '**if**' true then the statement inside this is executed.
**elif** -> is used to check multiple condition '**if**' false then next it will go to next '**elif**' if mentioned or to '**else**' statement.
**else** -> when all the mentioned condition are false then else gets executed.

We can multiple '**elif**' or '**if**' statement.

Q22. Write a code to take the age of person as an input and if age >= 18 display "I can vote". If age is < 18 display "I can't vote".

➔
```
age = int(input())
if age >= 18:
   print("I can vote")
else:
   print("I can't vote")
```

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

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

➔
```
numbers = [12, 75, 150, 180, 145, 525, 50]
sum = 0
for i in numbers:
  if i % 2 == 0:
```

```
        sum = sum + i
    print(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("Enter first number = "))
num2 = int(input("Enter second number = "))
num3 = int(input("Enter third number = "))
if num1>num2 and num1>num3:
    print(num1)
elif num2>num1 and num2>num3:
    print(num2)
else:
    print(num3)
```

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

- The number must be divisible by five

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

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

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

```
numbers = [12, 75, 150, 180, 145, 525, 50]
for i in numbers:
    if i>500:
        break
    elif i%5==0 and i<=150:
        print(i)
```

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

➔ Strings are arrays of bytes representing Unicode character.

```
Declaration:
Str1 = “iNeuron”
Str2 = ‘iNeuron’
Str3=’’’ iNeuron
         iNeuron
         iNeuron’’’
```

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

➔ Str1 = “iNeuron”

```
C1 = Str1[0] -> i
```

We can access the string using indexing.

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"
c1 = string[9:]
print(c1[-1::-1])
```

Q30. Reverse the string given in the above question.

➔
```
string = "Big Data iNeuron"
print(string[-1::-1])
```

Q31. How can you delete entire string at once?

➔ Using del keyword.

Q32. What is escape sequence?

➔ Escape sequence is a sequence of character which is used inside a character or string and doesn't represent itself but it get converted into character or series of character. Example - \' -> single quote, \t -> tab, etc.

Q33. How can you print the below string?

```
'iNeuron's Big Data Course'
```

➔
```
print('iNeuron\'s Big Data Course')
```

Q34. What is a list in Python?

➔ List is a collection of values in which we can store different type of data. Like string, numeric. List is represented like List1 = [ ].

Q35. How can you create a list in Python?

➔ List1= [ ] by using square bracket.

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

➔ By using indexing.

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

```
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.

➔
```
lst = [1,2,3,"Hi",[45,54, "iNeuron"], "Big Data"]
print(len(lst))
```

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

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

➔
```
lst.insert(2,"Big")
print(lst)
```

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

➔ Tuple is represented using ( ).
Tuple is immutable where list is mutable.

Q41. How can you create a tuple in Python?

➔ Tuple1 = ( ) by using circle bracket.

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, it is not possible to add name after creating tuple due to its immutability.

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

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

➔
```
t1 = ("Welcome","to","iNeuron")
print(len(t1))
```

Q45. What are sets in Python?

➔ Set doesn't allow duplicate items.
Set elements are immutable but set as a whole is mutable.
Indexing is not supported in set.

Q46. How can you create a set?

➔
```
S1 = set( )
S1 = {0,1,2,3,4}
```

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

➔
```
S1 = set( )
S1.add("iNeuron")
```

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

➔ S1 = set( )
S1.add(“Welcome”)
S1.add("to")
S1.add("iNeuron")

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

➔ add() is used to add a single value.
update() is used to add multiple values.
add() is faster as compared with update().
add() accepts only immutable objects like tuple,int etc.
update() accepts iterable objects like lists.
update() do not accept single values which are not iterable like set1.update(5) //error.

Q50. What is clear() in sets?

➔ clear() is used to remove all elements from a set.
set1.clear()

Q51. What is a frozen set?

➔ frozenset() is an immutable version of set. whereas set is a mutable object.
frozenset([iterable])

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

➔ frozenset() is immutable whereas set is mutable.
No function will work with frozenset() like add() but in set it will work.

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

➔ union() will give all elements from sets but will exclude duplicate values.
set1 = {1,2,3,4}
set2 = {1,5}
set3 = set1.union(set2) # set3 = set1 | set2
print(set3) # {1,2,3,4,5}

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

➔ Intersection() returns a set that contains the similarity between two or more sets.

```
x = {'a','b','c'}
y = {'a','b','d'}
r = x.intersection(y)
 print(r)
```

Q55. What is dictionary in Python?

➔ Dictionary are used to store data in key value pair.
Key cannot be duplicate.
Value can repeat.
dic = {“name”: “Sourav”, “Age”: 24}

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

➔ In dictionary we store data in form of key value pair but in other this concept is not there.

Q57. How can we declare a dictionary in Python?

➔ dic = {“name”: “Sourav”, “Age”: 24}

Q58. What will the output of the following?

```
var = {}
print(type(var))
```

➔ dictionary

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

```
➔ var = {}
➔ var["name"] ="Sourav"
➔ var["age"] =24
➔ print(var)
```

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

```
➔ var = {}
➔ var["name"] ="Sourav"
➔ var["age"] =24
➔ var["city"] ="Delhi"
➔ for k,v in var.items():
➔     print("Key is",k, "and value is",v)
```

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

```
➔ var={"name":"Sourav","age":"24","city":{"city1":"Delhi","city2":"mumbai
  "}}
➔ print(var["city"]["city1"])
➔ print(var["city"]["city2"])
```

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

➔ The get() is used to return the value with specified key.

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

➔ The items() return a view object that display a dictionary, as tuple in a list.

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

➔ The pop() is used to remove element from list.

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

➔ It remove the element that is last inserted in a dictionary as a tuple.

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

➔ Key() returns the keys in a dictionary as a list.

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

➔ Values() return the values in a dictionary as a list.

Q68. What are loops in Python?

➔ Loops in python means iterating over an instruction till condition is meet.

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

➔ There are 2 types of loop in python – for and while loop.

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

➔ We use for loop when we know the no of iteration and we use while loop when we are not
➔ aware of no of iteration while loop execute till the condition is true.

Q71. What is the use of continue statement?

➔ Continue will skip the current iteration and will move to next iteration.

Q72. What is the use of break statement?

➔ Break statement will stop and come out of the current iteration.

Q73. What is the use of pass statement?

➔ Pass is used when we want don’t want to write any code inside any function, if etc and to avoid getting error.

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

➔ range() will provide a sequence of number in a given range.
➔ range(5) – 0,1,2,3,4.

Q75. How can you loop over a dictionary?

➔ `for k,v in dic.items():`
➔ `dic is a dictionary.`

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

```
➔ n = 5
➔ fact = 1
➔ for i in range(1,n+1):
➔     fact = fact*i
➔ print(fact)
```

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

```
➔ p = int(input("Enter the principal ="))
➔ r = int(input("Enter the rate ="))
➔ t = int(input("Enter the time ="))
➔ SI = (p*r*t)/100
➔ print(SI)
```

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

```
➔ p = int(input("Enter the principal ="))
➔ r = int(input("Enter the rate ="))
➔ t = int(input("Enter the time ="))
➔ A = p*(1+r/100)**t
➔ print(A)
```

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

```
➔ n = 5
➔ c = 0
➔ for i in range(1,n+1):
➔     if n % i == 0:
➔         c=c+1
➔ if c==2:
➔     print("Prime Number")
➔ else:
➔     print("Not a Prime Number")
```

Q80. Write a Python program to check Armstrong Number.

```
➔ num= int(input())
➔ num2=num
➔ res = int(0)
```

```
n = len(str(num))
while num>0:
    d=num%10
    res=res+(d**n)
    num=num//10
print(res)
if res == num2:
    print("Armstrong number")
else:
    print("Not an Armstrong number")
```

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

```
nt = int(input())
n1, n2 = 0, 1
count = 0
if nt <= 0:
   print("Please enter a positive integer")
elif nt == 1:
   print("Fibonacci sequence upto",nterms,":")
   print(n1)
else:
   print("Fibonacci sequence:")
   while count < nt:
       print(n1)
       nth = n1 + n2
       n1 = n2
       n2 = nth
       count += 1
```

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

```
list1 = [1,2,3,4,5]
l = len(list1)
a = list1[0]
list1[0]=list1[l-1]
list1[l-1]=a
print(list1)
```

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

```
l = [1,2,3,4]
a=l[0]
l[0]=l[1]
l[1]=a
print(l)
```

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

```
➔ l = [1,2,3,4,5]
➔ print(max(l))
```

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

```
➔ list1 = [1,2,3,4,5]
➔ list2 = []
➔ s=0
➔ l = len(list1)
➔ for i in range(l):
➔     s=s+list1[i]
➔     list2.append(s)
➔ print(list2) #[1, 3, 6, 10, 15]
```

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

```
➔ str1 = input()
➔ str2 = str1[::-1]
➔ if (str1 == str2):
➔     print("String is palindrome")
➔ else:
➔     print("String is not palindrome")
```

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

```
➔ str1 = input("Enter the String = ")
➔ ith = int(input("Enter the i'th element to remove = "))
➔ l = []
➔ str2 = ' '
➔ leng = len(str1)
➔ for i in range(leng):
➔     if i == ith:
➔         continue
➔     else:
➔         l.append(str1[i])
➔ for i in l:
➔     str2 = str2 + i
➔ print(str2)
```

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

```
➔ str1 = "Hello, how are you"
➔ str2 = input()
➔ if str2 in str1:
➔     print("Yes substring is present");
```

```
➔ else:
➔     print("No substring is not present")
```

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

```
➔ str1 = "Welcome to iNeuron"
➔ k = int(input("Enter the value = "))
➔ l = len(str1)
➔ s = str1.split()
➔ for i in s:
➔     if k < len(i):
➔         print(i)
```

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

```
➔ dic = {1:[1,2],
➔        2:[3,4],
➔        3:[4,5]}
➔ l = list(dic.values())
➔ l1 = []
➔ for i in l:
➔     l1.extend(i)
➔ s = set(l1)
➔ print(s)
```

Q91. Write a Python program to merge two dictionary.

```
➔ dic1 = {1:1,2:2,3:3}
➔ dic2 = {4:4,5:5,6:6}
➔ dic3 = dic2.copy()
➔ dic1.update(dic3)
➔ print(dic1)
```

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}
```

```
➔ inp = [('Sachin', 10), ('MSD', 7), ('Kohli', 18), ('Rohit', 45)]
➔ output = {}
➔ for i in inp:
➔     output[i[0]] = i[1]
➔ print(output)
```

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)]
```

```
➔ l = [9, 5, 6]
➔ list_cube = [(i,i**3) for i in l]
➔ print(list_cube)
```

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)]
```

```
➔ tuple1 = (7, 2)
➔ tuple2 = (7, 8)
➔ tuple_comb = [(t1,t2) for t1 in tuple1 for t2 in tuple2 ] + [(t1,t2)
   for t1 in tuple2 for t2 in tuple1]
➔ print(tuple_comb)
➔
```

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)]
```

```
➔ l1 = [('for', 24), ('Geeks', 8), ('Geeks', 30)]
➔ print(sorted(l1,key=lambda x: x[1]))
```

Q96. Write a python program to print below pattern.

```
*
* *
* * *
* * * *
* * * * *
```

```
➔ n = 5
➔ for i in range(0,5):
➔     for j in range(0,i+1):
➔         print("*", end =" ")
➔     print("\n")
```

Q97. Write a python program to print below pattern.

```
    *
   **
  ***
 ****
*****
```

```
➔ n = 5
➔ i=0
➔ while(i<=5):
➔     print((n-i) * " " + i * "*")
➔     i=i+1
```

Q98. Write a python program to print below pattern.

```
    *
   * *
  * * *
 * * * *
* * * * *
```

```
➔ n = 5
➔ k=n-1
➔ for i in range(0,5):
➔     for j in range(0,k):
➔         print(end =" ")
➔     k = k-1
➔     for j in range(0,i+1):
➔         print("*",end = " ")
➔     print("\r")
```

Q99. Write a python program to print below pattern.

```
1
1 2
1 2 3
1 2 3 4
1 2 3 4 5
```

```
➔
➔ for i in range(0,5):
➔     k = 1
➔     for j in range(0,i+1):
➔         print(k,end =" ")
➔         k = k+1
➔     print("\r")
```

Q100. Write a python program to print below pattern.

```
A
B B
C C C
D D D D
E E E E E
```

```
➔ k = 65
➔ for i in range(0,5):
➔     for j in range(0,i+1):
➔         print(chr(k),end =" ")
➔     k = k + 1
➔     print("\r")
```