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

Ans: A string is a immutable data types in python. We create it by enclosing a character or sequence of characters into single, double, or triple quotations.

Immutable: The data types which can not be changed after their creation are called immutable data types.

Example:

```
# How to create a string in python

a="I have joined Big Data bootcamp"
print(a)
```

```
I have joined Big Data bootcamp
```

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

Ans: The characters in string can be accessed by specifying string name followed by an index number in square brackets []. The indexing in string is start from 0. So, we can say that string in python has 0 based indexing.

Example:

```
# How to access the characters in string in python

a="iNeuron"
print(a[0])

# To know the totl length of a string we will code like

len(a)

# To reverse a string

print(a[::-1])

# To acces range of characters in string, the slicing method will be used

print(a[0:5])
```

```
i
norueNi
iNeur
```

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

```
string = "Big Data iNeuron"
desired_output = "iNeuron"
```

Ans:

```
#string = "Big Data iNeuron"
#desired_output = "iNeuron"


a="Big Data iNeuron"
print(a[8:])
```

```
iNeuron
```

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

```
string = "Big Data iNeuron"
desired_output = "norueNi"
```

Ans:

```
a= "Big Data iNeuron"
print(a[-1:-8:-1])
```

```
norueNi
```

Q30. Resverse the string given in the above question.

Ans:

```
#string = "Big Data iNeuron"
#desired_output = "norueNi"


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

```
norueNi ataD giB
```

Q31. How can you delete entire string at once?

Ans: We use the del statement to delete an entire string.

Example:

```
#string = "Big Data iNeuron"
#desired_output = "norueNi"


a="Big Data iNeuron"
del a
```

Q32. What is escape sequence?

Ans: The escape sequence allows to insert special characters in the string because string itself does not allow to insert special characters. In this way we add backslash followed by the character that we want to insert.

Q33. How can you print the below string?

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

Ans: We will use the escape sequence to print this string.

```
a="iNeuron\'s Big Data Course"
print(a)
```

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

Q34. What is a list in Python?

Ans: The list is a mutable data type in python. I can have any type of elements such as numbers, string, tuples, dictionary, and sets.

Q35. How can you create a list in Python?

Ans: We can create a list in python by enclosing elements in the square brackets. Like as follows.

```
a=[1,2,3,3.4,”iNeuron”]
```

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

Ans: We can access the elements in the list by giving name of the list followed by the index number in the square brackets.

Example:

```
a=[1,2,3,4,22.3,"iNeuron"]
print(a[0])
print(a[1])
print(a[2])
print(a[3])
print(a[5])
```

```
1
2
3
4
iNeuron
```

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

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

Ans: This is called nested list. Means list inside a list. So, we can access a specific element just as follows:

```
lst = [1,2,3,"Hi",[45,54, "iNeuron"], "Big Data"]
print(lst[4][2])
```

```
iNeuron
```

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

Ans:

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

```
6
```

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

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

Ans:

```
lst = ["Welcome", "to", "Data", "course"]
lst.insert(2,"Big")
print(lst)
```

```
['Welcome', 'to', 'Big', 'Data', 'course']
```

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

Ans: The tuple is an immutable and ordered data type in python. The duplicate elements in tuple data type may exist.

Q41. How can you create a tuple in Python?

Ans: We can create tuple in python by putting comma separated values. we can put comma separated values between parentheses () also.

```
tup1="A","B","C","D","E","F","G","H",1,2,3
print(tup1)

# We also can putt comma separated valuse betqweeen the paratheses

tup2=('A', 'B', 'C', 'D', 'E', 'F', 'G', 'H', 1, 2, 3)
print(tup2)
```

```
('A', 'B', 'C', 'D', 'E', 'F', 'G', 'H', 1, 2, 3)
('A', 'B', 'C', 'D', 'E', 'F', 'G', 'H', 1, 2, 3)
```

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: we cannot add the name to tuple because tuple is a immutable data type. So, no element in tuple can be changed, update or remove. Tuple only represents the data that don't need to be update.

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

Ans: Yes, two tuples can be appended. We can combine two tuples by concatenation.

```
tup1="A","B","C","D","E","F","G","H",1,2,3
tup2=('I', 'J', 'K', 'L', 'M', 'N', 'O', 'P', 4, 5, 6)
tup3=tup1 + tup2
print(tup3)
```

```
('A', 'B', 'C', 'D', 'E', 'F', 'G', 'H', 1, 2, 3, 'I', 'J', 'K', 'L', 'M', 'N', 'O', 'P', 4, 5, 6)
```

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

Ans:

```
tup1="A","B","C","D","E","F","G","H",1,2,3
print("The tuple is ",tup1)
print("Total elements in tuple are ",len(tup1))
```

```
The tuple is  ('A', 'B', 'C', 'D', 'E', 'F', 'G', 'H', 1, 2, 3)
Total elements in tuple are  11
```

Q45. What are sets in Python?

Ans: The set is a mutable data type in python. It can be unordered. It has only those values which are belong to the immutable data types only. Just like numbers, strings, and tuples. **It do not support indexing.**

Q46. How can you create a set?

Ans: In python we can create set by putting comma separated values between the curly brasses. Just as follows:

```
thisset = {"apple", "banana", "grapes",2, 4,12.333,10>5}
print(type(thisset))
print(thisset)
```

```
<class 'set'>
{True, 2, 4, 12.333, 'banana', 'grapes', 'apple'}
```

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

Ans: we can add a new element in the set just as follows:

```
thisset = {"apple", "banana", "grapes",2, 4,12.333,10>5}
print(type(thisset))
thisset.add('iNeuron')
print(thisset)
```

```
<class 'set'>
{True, 2, 4, 12.333, 'banana', 'iNeuron', 'grapes', 'apple'}
```

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

Ans:

```
thisset = {"apple", "banana", "grapes",2, 4,12.333,10>5}
thisset.add('iNeuron')
thisset.add('BigData')
thisset.add('ML')
thisset.add('150')
print(thisset)
```

```
{True, 2, 'BigData', 4, 12.333, 'banana', 'iNeuron', 'ML', '150', 'grapes', 'apple'}
```

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

Ans: The add( ) function is used to add a single value in the set. While update() function allows to add sequence of values in the set. Furthermore, the add() function is accept only immutable arguments such as integers, strings, tuple etc. While update() function is only accepts iterable sequences.

Example:

```
# Adding multiple elements in set via add() function
thisset = {"apple", "banana", "grapes",2, 4,12.333,10>5}
thisset.add('iNeuron')
thisset.add('BigData')
thisset.add('ML')
thisset.add(150)
print(thisset)

# Now, adding elements via update() function
myset = {"apple", "banana", "grapes",2, 4,12.333,10>5}
myset.update([2,3,4,5,6])
print(myset)
```

```
{True, 2, 'BigData', 4, 12.333, 'banana', 'iNeuron', 'ML', 150, 'grapes', 'apple'}
{True, 2, 3, 4, 5, 6, 12.333, 'banana', 'grapes', 'apple'}
```

Q50. What is clear() in sets?

Ans: The clear() function in a set used to remove all the elements from the set.

```
thisset = {"apple", "banana", "grapes",2, 4,12.333,10>5}
thisset.clear()
print(thisset)
```

```
set()
```

Q51. What is frozen set?

Ans: Frozen set is nothing but a immutable version of the set. It return the value that cannot be changed.

Q52. How is frozen set different from set?

Ans: As we know that the typical set is a mutable data type. We can add or update values in it. While frozen set is a set where we cannot do add or update process. This is the main difference between set and frozen set.

Example:

```
# We can add valuse in set as they are mutable
thisset = {"apple", "banana", "grapes",2, 4,12.333,10>5}
thisset.add('Hello!')
print(thisset)

# We cannot add or updata frozen set as they are immutable version of set
# Our program will give the error when we will try to do so
myset={"apple", "banana", "grapes",2, 4,12.333,10>5}
a=frozenset(myset)
a.add('Hello!')
print(a)
```

```
{True, 2, 4, 12.333, 'banana', 'grapes', 'Hello!', 'apple'}

---------------------------------------------------------------------------
AttributeError                            Traceback (most recent call last)
Input In [134], in <cell line: 9>()
      7 myset={"apple", "banana", "grapes",2, 4,12.333,10>5}
      8 a=frozenset(myset)
----> 9 a.add('Hello!')
     10 print(a)

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

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

Ans: The union() function returns all the elements from the original set as well as all the elements from the specified set(s). We can specify as many sets as we want.

Example:

```
x={'a','c','d'}
y={'e','f','g','h'}
z={'i','j','k','l'}
d={1,2,3,4}
a=x.union(y,z,d)
print(a)
```

```
{'e', 'k', 1, 2, 3, 'f', 4, 'a', 'j', 'h', 'd', 'l', 'g', 'i', 'c'}
```

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

Ans: The intersection is nothing but return the common values of two or more sets.

Example:

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

```
{'d', 'a'}
```

Q55. What is dictionary in Python?

Ans: The dictionaries used to store the values in key:value pairs. They are ordered (In python 3.7). They are mutable (only values not keys, keys must be unique) and written in the curly braces.

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

Ans: The main difference between dictionary and all other data types is that values in dictionary are written in key:value pair which is not the case for other data types.

Q57. How can we delare a dictionary in Python?

Ans: We can create a dictionary in python just by putting values separated by comma inside curly braces.

Example:

```
my_dict={1:'iNeuron',2:'BigData',3:'Bootcamp',4:[1,2,3,4]}
print(my_dict[4])

# We also can add string as a key, just as
my_dict1={'Qamar':'Data Engr.', 'Hassan':'ML Engr.'}
print(my_dict1['Hassan'])
```

```
[1, 2, 3, 4]
ML Engr.
```

Q58. What will the output of the following?

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

Ans:

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

```
<class 'dict'>
```

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

Ans: We can add or change the elements in dictionary with the help of assignment operator. If the key is already present in the dictionary, then the new value will be assigned to that key otherwise we also can add new value and key to the dictionary.

Example:

```
# To add a new value in the dictionary
my_dict={1:'Yasir', 2:'Ali', 3:'Muzammil', 4:'Taimoor'}
print(my_dict)
my_dict[5]='Dawood'
print(my_dict)

# To change the value in an existing dictionary
my_dict[5]='Qamar'
print(my_dict)
```

```
{1: 'Yasir', 2: 'Ali', 3: 'Muzammil', 4: 'Taimoor'}
{1: 'Yasir', 2: 'Ali', 3: 'Muzammil', 4: 'Taimoor', 5: 'Dawood'}
{1: 'Yasir', 2: 'Ali', 3: 'Muzammil', 4: 'Taimoor', 5: 'Qamar'}
```

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

Ans:

```
my_dict={1:'Yasir', 2:'Ali', 3:'Muzammil', 4:'Taimoor'}
print(my_dict.values())

# We also can access all the keys in our dictionary
print(my_dict.keys())
```

```
dict_values(['Yasir', 'Ali', 'Muzammil', 'Taimoor'])
dict_keys([1, 2, 3, 4])
```

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

Ans:

```
my_dict={'Name':'Qamar','Grades':{'Mathematics':'100','English':'90','Chemistry':'80'},'Nationality':'Pakistani'}
print(my_dict['Grades'])
```

```
{'Mathematics': '100', 'English': '90', 'Chemistry': '80'}
```

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

Ans: The get() function is used to print the value of specified key if present in the dictionary. The syntax of get() function is "dictionary.get(key, value)". Here key is mandatory part while value is optional. There will be three possibilities then.

1: if the key is given in dictionary then get() method will print the value of that key.

2: if the key is not given in the dictionary, but we pass the value in get(), it will print that value.

3: if both are not present then the get() will return none.

Example:

```
# If the key is given in the dictionary
my_dict={'Name':'Qamar','Grades':{'Mathematics':'100','English':'90','Chemistry':'80'},'Nationality':'Pakistani',}
print(my_dict.get('Name'))

# If the key is not given then we can pass value to the get() function to get that value instead of none
print(my_dict.get('i', 'iNeuron'))

# If both the arguments are not given in the dictionary, it will return none
print(my_dict.get(4))
```

```
Qamar
iNeuron
None
```

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

Ans: The items() function print the list of dictionary's key value pairs. It does not take any argument.

Example:

```
# If the key is given in the dictionary
my_dict={'Name':'Qamar','Grades':{'Mathematics':'100','English':'90','Chemistry':'80'},'Nationality':'Pakistani',}
print(my_dict.items())
```

```
dict_items([('Name', 'Qamar'), ('Grades', {'Mathematics': '100', 'English': '90', 'Chemistry': '80'}), ('Nationality', 'Pakista
ni')])
```

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

Ans: The pop() function is used to remove a element of a specific index from a list. To do so, we will pass a index as a argument in the pop() method to remove the value from that index (If we do not pass the argument then by default it will take -1 which will be the last value). It also can be used to print the removed element or values.

Example:

```
mylist=['Apple','Banana','Orange','Mango']
mylist.pop(1)
print(mylist)

# We can print the removed value too, using pop() function
mylist=['Apple','Banana','Orange','Mango']
x=mylist.pop(1)
print(x)
```

```
['Apple', 'Orange', 'Mango']
Banana
```

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

Ans: The popitem() is used to remove the last inserted item from a dictionary. The return value of the popitem() method will be the last removed item of that dictionary.

Example:

```
my_dict={1:"Qamar",2:"Yasir",3:"Ali"}
my_dict.popitem()
```

```
(3, 'Ali')
```

Q68. What are loops in Python?

Ans: The loops allowed to execute a statement or group of statement multiple times. There are three looping structures in python. While, For and Nested Loop.

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

Ans: There are three types of loops in python. While, For and Nested Loop.

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

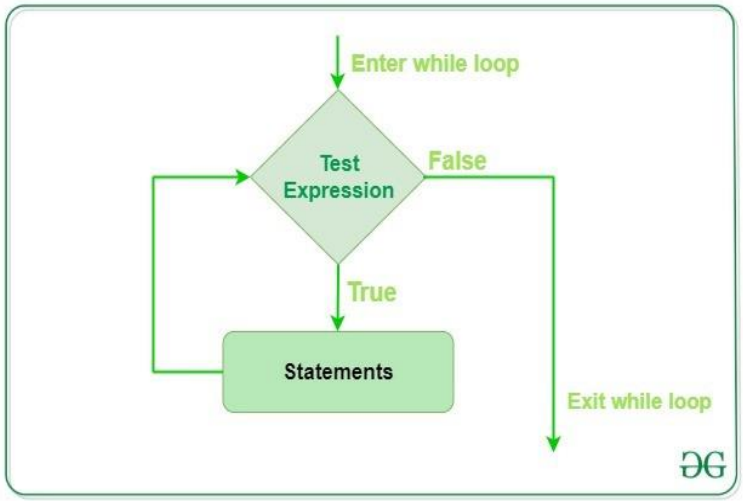
Ans: While Loop; The while loop repeats the statement or group of statements until the given condition remains true. (It is also called infinite loop. In such case it repeats the statements infinite number of times until the external factor stop it. It happens when the given condition never false). The While loop is used when the number of iterations are unknown (Iteration means repeated execution of a set of statements).

Example:

```
a=1
while a<=5:
    print(a)
    a=a+1
```

```
1
2
3
4
5
```

Structure of While Loop:



For Loop: The For Loop repeats a statemen or set of statements for fixed number of time. It is used when the number of iterations are known.

Syntax:

```
For <Variable> in <range>:
    Statements
```

Here: Range (Start, Stop, Step)

Start: Starting number of sequences

Stop: Ending number of sequences

Step: Difference between the numbers

Note: The range() function never include the ending number (stop number).

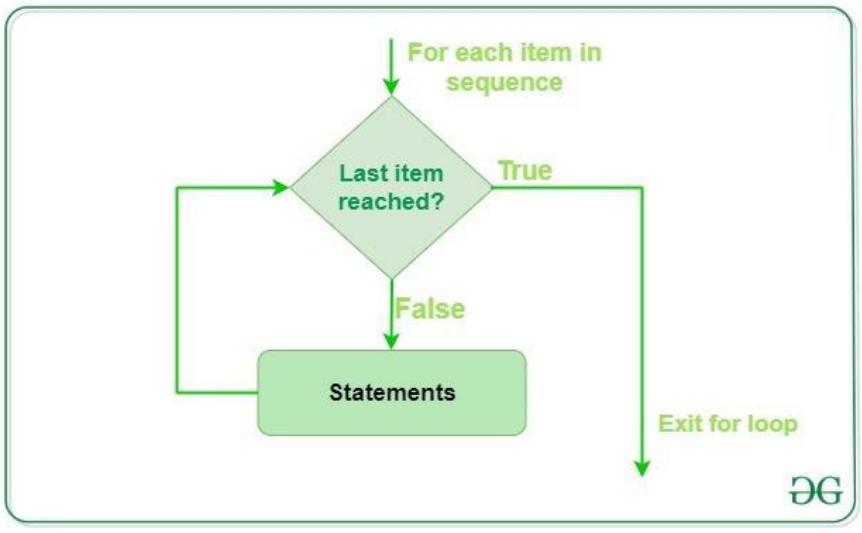
Example:

```
for i in range(1, 10, 2):
    print(i)
```

```
1
3
5
7
9
```

We can see in the example that range() function did not include the ending number (Stop).

Structure of For Loop:



Q71. What is the use of continue statement?

Ans: The continue statement is used to end the current iteration in For and While Loop and starts the next iteration. Always put semicolon after continue keyword.

Example:

```
a=0
while a<9:
    a=a+1
    if a==3:
        continue;
    print(a)

# Use of contunue statement with the range() function
# We use this wehn we have to print sequence of numbers
# without print a specific one (and we are bound to do it with loop)
# and only have to use the single loop

print("Sequence of numbers without 5 is as follows!")
for i in range(1, 10):
    if i==5:
        continue;
    print(i)
```

```
1
2
4
5
6
7
8
9
Sequence of numbers without 5 is as follows!
1
2
3
4
6
7
8
9
```

Q72. What is the use of break statement?

Ans: The break statement used to terminate the execution of a loop. It put inside the body of the loop (Generally after the if condition). It terminates the loop in which it appears and execute the next statement (if any) immediately. In case of nested loop it terminates the inner most loop.

Example:

```
for i in range(10):
    print(i)
    if i==5:
        break
```

```
0
1
2
3
4
5
```

Q73. What is the use of pass statement?

Ans: Pass statement is used as a place holder to code in future. When a programmer does not want to write a code then he places the pass statement where empty code is not allowed like in if statements, loops, functions definition, or class definition. So, using pass statement user avoid the errors.

Example:

```
# Pass statement while working wuith loops
for i in range(1, 10):
    pass

# Pass statement while working with if statements.
a=10
b=20
if a<b:
    pass
else:
    print("a is greater")

# Pass statement while defining class.
class iNeuron:
    pass

# Pass statement while defining functions.
def function:
    pass
```

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

Ans: The range function used to print the sequence of numbers, starting from 0 (By default, we also can specify the number of our choice), increments by 1 (By default, here we also can specify number of our choice) and stops before the specified number.

Example:

```
for i in range(1, 10, 2):
    print(i)

# If we do not set the starting and ending points then,

print("The numbers are as follows!")
for n in range(5):
    print(n)
```

```
1
3
5
7
9
The numbers are as follows!
0
1
2
3
4
```

Q75. How can you loop over a dictionary?

Ans: We can loop over a dictionary in different ways to access keys or values and key value pair as well.

Example:

```
# Access all the keys without using key(), that are present in dictionary
mydict={1:"QAMAR", 2:"WAQAR", 3:"ALI", 4:"YASIR", 5:"DAWOOD"}
print("The keys in the dictionaary are:")
for keys in mydict:
    print(keys)

# Iterate through all the values using .value(), that are present in dictionary
print("The values in the dictionary are:")
for values in mydict.values():
    print(values)

# Iterate through all the keys, and vlue pairs using items()
print("The key:value pairs in the dictionayr are:")
for keys, values in mydict.items():
    print(keys,":", values)
```

```
The keys in the dictionaary are:
1
2
3
4
5
The values in the dictionary are:
QAMAR
WAQAR
ALI
YASIR
DAWOOD
The key:value pairs in the dictionayr are:
1 : QAMAR
2 : WAQAR
3 : ALI
4 : YASIR
5 : DAWOOD
```

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 not exist for 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)
```

```
Enter a number: 3
The factorial of 3 is 6
```

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

Ans:

```
#A = P(1+ R/100)^t.

p=int(input("Please enter the principle amount = "))
if p==0 or p<0:
    raise Exception("You are entering invalid amount")
r=float(input("Please enter the interest rate = "))
if r==0 or r<0:
    raise Exception("You are entering invalid interest rate")
t=int(input("Please enter the time duration = "))
if t==0 or t<0:
    raise Exception("You are entering invalid value")
x=(p*r*t)/100
print("The simple interest is",x)
```

```
Please enter the principle amount = 1000
Please enter the interest rate = 2.22
Please enter the time duration = 1
The simple interest is 22.2
```

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

Ans:

```
#A = P(1+ R/100)^t.

p=int(input("Please enter the principle amount = "))
if p==0 or p<0:
    raise Exception("You are entering invalid amount")
r=float(input("Please enter the interest rate = "))
if r==0 or r<0:
    raise Exception("You are entering invalid interest rate")
t=int(input("Please enter the time duration = "))
if t==0 or t<0:
    raise Exception("You are entering invalid value")
x=p*(1+r/100)**t
print("The compound interest is",x)
```

```
Please enter the principle amount = 1000
Please enter the interest rate = 2.4567
Please enter the time duration = 2
The compound interest is 1049.737537489
```

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

Ans:

```
number = int(input("Enter The Number"))
if number > 1:
    for i in range(2,int(number/2)+1):
        if (number % i == 0):
            print(number, "is not a Prime Number")
            break
    else:
        print(number,"is a Prime number")
# If the number is less than 1 it can't be Prime
else:
    print(number,"is not a Prime number")
```

```
Enter The Number4
4 is not a Prime Number
```

Q80. Write a Python program to check Armstrong Number.

Ans:

```
n = int(input("Enter s number"))
s = n
b = len(str(n))
sum1 = 0
while n != 0:
    r = n % 10
    sum1 = sum1+(r**b)
    n = n//10
if s == sum1:
    print("The given number", s, "is armstrong number")
else:
    print("The given number", s, "is not armstrong number")
```

```
Enter s number999
The given number 999 is not armstrong number
```

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

Ans: a series of numbers in which each number (*Fibonacci number*) is the sum of the two preceding numbers. The simplest is the series 1, 1, 2, 3, 5, 8, etc. it can be defined as
$F_n = F_{n-1} + F_{n-2}$

Example:

```
def Fibonacci(n):
    if n<=0:
        return ("Invalid input")
    elif n==1:
        return 0
    elif n==2:
        return 1
    else:
        return Fibonacci(n-1)+Fibonacci(n-2)
print(Fibonacci(20))
```

```
4181
```

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

Ans:

```
def myList(swap):
    swap[0], swap[-1] = swap[-1], swap[0]
    return swap
swap=[1,2,3,4,5,6,7,8,9,10]
print(myList(swap))
```

```
[10, 2, 3, 4, 5, 6, 7, 8, 9, 1]
```

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

Ans:

```
def myList(swap):
# I'll swap 2 elements at the indexes 2 and 3 with elements at -2 and -3
    swap[2], swap[3], swap[-2], swap[-3] = swap[-2], swap[-3], swap[2], swap[3]
    return swap
swap=[1,2,3,4,5,6,7,8,9,10]
print(myList(swap))
```

```
[1, 2, 9, 8, 5, 6, 7, 4, 3, 10]
```

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

Ans:

```
n=int(input("Enter the N value"))
list=[666,1000,44,556,765,20000]
list.sort()
print(list[-n:])
```

```
Enter the N value2
[1000, 20000]
```

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

Ans: Cumulative sum means total sum of data grows with time.

```
list=[10, 20, 30, 40, 66, 78,99, 100]
list1=[]
j=0
for i in range(0, len(list)):

    j+=list[i]
    list1.append(j)
print(list1)
```

```
[10, 30, 60, 100, 166, 244, 343, 443]
```

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

Ans: A string is called palindrome if the string reading from left to right is equal to reading from right to left.

Example:

```
x = "LEVEL"

w = ""
for i in x:
    w = i + w

if (x == w):
    print("Yes, The given string is plindrome")
else:
    print("No")
```

```
Yes, The given string is plindrome
```

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

Ans:

```
# Initializing String
test_str = "MuhammadQamarAmin"

# Removing char at pos 3
# using replace
new_str = test_str.replace('a', '')

# Printing string after removal
# removes all occurrences of 'a'
print ("The string after removal of i'th character( doesn't work) : " + new_str)

# Removing 1st occurrence of s, i.e 5th pos.
# if we wish to remove it.
new_str = test_str.replace('m', '', 1)

# Printing string after removal
# removes first occurrences of s
print ("The string after removal of i'th character(works) : " + new_str)
```

```
The string after removal of i'th character( doesn't work) : MuhmmdQmrAmin
The string after removal of i'th character(works) : MuhamadQamarAmin
```

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

Ans:

```
# There are different methods to check
# whether the substring is present in the given
# string or not but we'll use "in" operator

a=input("Enter the substring that you want to check: ")
mystr="I am learning Big Data from iNeuron"
if a in mystr:
    print("Yes! It is present in the given string")
else:
    print("No! It is not present in the given list")
```

```
Enter the substring that you want to check: iNeuron
Yes! It is present in the given string
```

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

Ans: We will use the list comprehension to find the words which are greater than given length. It is a method to create the list from string or from another list. The syntax for the list comprehension is **`[expression for element in iterable if condition]`**

Program:

```
# Program to find all the words which are greater than given length k
# using the list comprehension

a=int(input("Enter the value of K! "))
mystr=("I am a stuent of MS Data Science at FAU")
print([word for word in mystr.split() if len(word)==a])

# split() method splits a strig in the list
```

```
Enter the value of K! 4
['Data']
```

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

Ans:

```
mydict={'a':[1,2,3,4], 'b':[3,4,6,7], 'c':[7,8,9,10]}
print("The original dictionary is! ", mydict)

# Now, to extract the unique dictionary values we'll use the
# list comprehension, sorted() and values() method

a=list(sorted({a for val in mydict.values() for a in val}))
print("the list of the unique values is! ",a)
```

```
The original dictionary is!  {'a': [1, 2, 3, 4], 'b': [3, 4, 6, 7], 'c': [7, 8, 9, 10]}
the list of the unique values is!  [1, 2, 3, 4, 6, 7, 8, 9, 10]
```

Q91. Write a Python program to merge two dictionaries.

Ans:

```
# We'll use the unpack operator (**) to merge two dictionaries

a={'a':[1,2], 'b':[3,4]}
b={'c':[2,3]}
x={**a,**b}
print("After the merge, the dictionary is ", x)
```

```
After the merge, the dictionary is  {'a': [1, 2], 'b': [3, 4], 'c': [2, 3]}
```

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: The simplest method to convert the list or tuple into dictionary, we pass a tuple into a dict()

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

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

We also can do it with the help of setdefault() method. In this method, we have used the dictionary method *setdefault()* to convert the first parameter to key and the second to the value of the dictionary. *setdefault(key, def_value)* function searches for a key and displays its value and creates a new key with def_value if the key is not present. Using the append function we just added the values to the dictionary.

```
list1=[('Sachin', 10), ('MSD', 7), ('Kohli', 18), ('Rohit', 45)]
dict1=dict()
for students, numbers in list1:
    dict1.setdefault(students, []).append(numbers)
print(dict1)
```

```
{'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)]
```

Ans: We can use the pow() function with list comprehension to create the list of tuples. The first element will simply be a element and the second will be cube of that number.

```
list = [9, 5, 6]
z=[(val, pow(val, 3)) for val in list]
print("Tuple list of the given list is ",z)
```

```
Tuple list of the given list is  [(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)]
```

Ans: We will solve it using chain() and product() method. In this, we perform task of pair creation using product(), and chain() is used to add both the results from product() used twice.

```
# Importing chain and product from itertools

from itertools import chain, product
test_tuple1 = (4, 5)
test_tuple2 = (7, 8)

# printing original tuples
print("The original tuple 1 : ",test_tuple1)
print("The original tuple 2 : " ,test_tuple2)

res1=list(chain(product(test_tuple1, test_tuple2), product(test_tuple2, test_tuple1)))

# printing result
print("The filtered tuple : ",res1)
```

```
The original tuple 1 :  (4, 5)
The original tuple 2 :  (7, 8)
The filtered tuple :  [(4, 7), (4, 8), (5, 7), (5, 8), (7, 4), (7, 5), (8, 4), (8, 5)]
```

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:

```
def Sort_Tuple(tup):
    return(sorted(tup, key = lambda x: x[1]))
tup = [('for', 24), ('Geeks', 8), ('Geeks', 30)]

# printing the sorted list of tuples
print(Sort_Tuple(tup))
```

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

Q96. Write a python program to print below pattern.

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

Ans:

```
for i in range(1,6):
    for j in range(i):
        print("*", end="")
    print("\n")
```

```
*

**

***

****

*****
```

Q97. Write a python program to print below pattern.

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

Ans:

```
for i in range(5):
    for s in range(-6, -i):
        print(" ", end="")
    for j in range(i+1):
        print("* ", end="")
    print()
```

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

Q98. Write a python program to print below pattern.

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

Ans:

```
for i in range(5):
    for s in range(-6, -i):
        print(" ", end="")
    for j in range(i+1):
        print("* ", end="")
    print()
```

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

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:

```
for i in range(6):
    for j in range(i):
        print("",j+1, end="")
    print("\n")
```

```
 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

Ans:

```
# Here we use the ord() function to get the asci values
# of the alphabets.
k=ord("A")
for i in range (6):
    for j in range(i):
# Here, the chr() method is to convert asci value to alphabet
        print(chr(k), end="")
        k=k+1
    print("\n")
```

```
A

BC

DEF

GHIJ

KLMNO
```