## Assignment Part-1

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

* Python can be used for creating variety of different programs not specific to any problems hence python is called General purpose.
* Python is using the English mnemonics like language so people can understand that easily

Q2. Why is Python called a dynamically typed language?

* Dynamic typing means that the type of the variable is determined only during runtime.

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

* Pros:
* Python is easy to learn and read
* Python has a vast collection of libraries
* Python is free, open-source, and has a vibrant community
* Python is a portable programming language
* Python is an interpreted language
* Cons:
  + Python has speed limitations
  + 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?

* Data Science, Machine Learning, Deep Learning, Artificial Intelligence, Scientific Computing Scripting, Networking, Game Development to Web Development

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

* Variables are used to store information to be referenced and used by programs
* In Python, we need not declare a variable with some specific data type. Python has no command for declaring a variable. A variable is created when some value is assigned to it. The value assigned to a variable determines the data type of that variable.

Thus, declaring a variable in Python is very simple.

* + Just name the variable
  + Assign the required value to it
  + The data type of the variable will be automatically determined from the value assigned, we need not define it explicitly

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

* + By using the input() function.
  + **input ():** This function first takes the input from the user and converts it into a string. The type of the returned object always will be <type ‘str’>. It does not evaluate the expression it just returns the complete statement as String.

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

* String

Q8. What is type casting?

* The conversion of one data type into the other data type is known as type casting
* For eg: There might be conditions when we might require integer input from the user/Console, so we can input(integer/float) from the console and typecasts them to an integer

num = int(input())

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

No. input() is a built in function to accept only one input.

Q10. What are keywords?

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

Eg: if, for, while, True, false, type

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

* We cannot use a keyword as a [variable](https://www.programiz.com/python-programming/variables-datatypes) name, [function](https://www.programiz.com/python-programming/function) name, or any other identifier. They are used to define the syntax and structure of the Python language.
* For eg ‘type()’ considered as function to know which class a variable belongs to.

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

* Indentation is adding a white space before a statement to a particular block of code.
* It tells the python interpreter that the group of statement belongs to particular block of code.

Q13. How can we throw some output in Python?

* Using the print() function

Q14. What are operators in Python?

* These are standard symbols used for the purpose of logical and arithmetic operations
* See the Arithmetic operator

| **Operator** | **Description** | **Syntax** |
| --- | --- | --- |
| + | Addition: adds two operands | x + y |
| – | Subtraction: subtracts two operands | x – y |
| \* | Multiplication: multiplies two operands | x \* y |
| / | Division (float): divides the first operand by the second | x / y |
| // | Division (floor): divides the first operand by the second | x // y |
| % | Modulus: returns the remainder when the first operand is divided by the second | x % y |
| \*\* | Power: Returns first raised to power second | x \*\* y |

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

* / is floating point division, and // is floor division (integer division)
* For eg: 5 / 2 returns 2.5 and 5 // 2 returns 2

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

```

iNeuroniNeuroniNeuroniNeuron

```

print("'''\niNeuroniNeuroniNeuroniNeuron\n'''")

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:'))  
if (num % 2) == 0:  
 print(num, "is even number")  
else:  
 print(num, "is odd number")

Q18. What are boolean operator?

* Boolean operators gives result of ‘True’ or ‘False’ after an operation

Comparison operators(>, <,>=,<=,==,!=)

and Logical operators( and, or, not)

Q19. What will the output of the following?

```

1 or 0

0 and 0

True and False and True

1 or 0 or 0

```

Q20. What are conditional statements in Python?

* Conditional Statement in Python perform different computations or actions depending on whether a specific Boolean constraint evaluates to true or false. Conditional statements are handled by IF statements in Python.

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

* If:- perform true or false condition of a statement
* Elif: - if the previous conditions were not true, then try this condition
* Else:- Perform anything which isn't caught by the preceding 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".

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]

```

#sum of all the even numbers  
  
numbers = [12, 75, 150, 180, 145, 525, 50]  
sum = 0  
for num in numbers:  
 if num % 2 == 0:  
 sum += num  
print("sum of numbers=", sum)

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

#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 first number:"))  
b = int(input("Enter the second number:"))  
c = int(input("Enter the third number:"))  
def max\_of\_numbers(a,b,c):  
 if (a >= b) and (a >= c):  
 greatest = a  
 elif (b >= c):  
 greatest = b  
 elif (c >= a):  
 greatest = c  
 return greatest  
  
max1= max\_of\_numbers(a, b, c)  
print('Greatest number is ', max1)

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 num in numbers:  
 if num > 500:  
 break  
 if num > 150:  
 continue  
 if num % 5 == 0:  
 print("The numbers divisible by 5 are:", num)

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

* A string is a data structure in Python that represents a sequence of characters. It is an immutable data type, meaning that once you have created a string, you cannot change it.
* We can declare strings by enclosing characters using single or double quotes or Triple quotes

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

* 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. The index of the last character will be the length of the string minus one.
* String indices can also be specified with negative numbers, in which case indexing occurs from the end of the string backward: -1 refers to the last character, -2 the second-to-last character, and so on

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[8:])

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[:8:-1])

Q30. Resverse the string given in the above question.

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

Q31. How can you delete entire string at once?

* Python will not allow a particular character in a string. Whereas we can remove the entire string variable using the ‘del’ command.

string = "Big Data iNeuron"  
del string  
print(string)

we will get the below error

NameError: name 'string' is not defined

Q32. What is escape sequence?

* 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.
* Escape characters can be classified as non-printable characters when backslash precedes them. The print statements do not print escape characters.

|  |  |
| --- | --- |
| **Code** | **Description** |
| \’ | Single quotation |
| \\ | Backslash |
| \n | New Line |
| \r | Carriage return |
| \t | Tab |
| \b | Backspace |
| \f | Form feed |
| \ooo | Octal equivalent |
| \xhhh | Hexadecimal equivalent |

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?

Lists are used to store multiple items in a single variable.

The following are the properties of a list.

* **Mutable:** The elements of the list can be modified. We can add or remove items to the list after it has been created.
* **Ordered:** The items in the lists are ordered. Each item has a unique index value. The new items will be added to the end of the list.
* **Heterogenous:** The list can contain different kinds of elements i.e; they can contain elements of string, integer, boolean, or any type.
* **Duplicates:** The list can contain duplicates i.e., lists can have two items with the same values.

Q35. How can you create a list in Python?

* Lists in Python can be created by just placing the sequence inside the square brackets[]

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

* In order to access the list items refer to the index number. Use the index operator [ ] to access an item in a list. The index must be an integer. Nested lists are accessed using nested indexing.

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.

user\_input = input(" enter the inputs with space:")  
lst = user\_input.split()  
print('length of list is:', len(lst))

enter the inputs with space: shiju 1 {3,4,5}

length of list is: 3

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(3,'Big')  
print(lst)

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

* Python Tuple is a collection of objects separated by commas. To create a tuple we will use () operators for eg: num = (1,5,20,’abc’)
* The key difference between the tuples and lists is that while the tuples are immutable objects the lists are mutable. This means that tuples cannot be changed while the lists can be modified. Tuples are more memory efficient than the lists.

Q41. How can you create a tuple in Python?

* To create a tuple we will use () operators for eg: num = (1,5,20,’abc’)

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

We cannot add an object into the tuple, because tuple is immutable.

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

There is no append function as such in tuple but we can append/concatenate two tuples by + operator.

# Code for concatenating 2 tuples  
  
tuple1 = (0, 1, 2, 3)  
tuple2 = ('python', 'course')  
  
# Concatenating above two  
print(tuple1 + tuple2)

(0, 1, 2, 3, 'python', 'course')

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

tuple1 = (0, 1, 2, 3)  
print(len(tuple1))

Q45. What are sets in Python?

* A Set is an unordered collection data type that is iterable, mutable, and has no duplicate elements.
* Set are represented by { } (values enclosed in curly braces)

Q46. How can you create a set?

* we can create sets by placing all the elements inside curly braces {}, separated by comma.
* A set can have any number of items and they may be of different types (integer, float, tuple, string etc.). But a set cannot have mutable elements like [lists](https://www.programiz.com/python-programming/list), sets or [dictionaries](https://www.programiz.com/python-programming/dictionary) as its elements.

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

set1 = {'a', 'b', 'c'}  
set1.add('iNeuron')  
print(set1)

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

set2 = {'a', 'b', 'c'}  
set2.add('e', 'f', 'g')  
print(set2)

getting the following error

TypeError: add() takes exactly one argument (3 given)

But we can add multiple values using iterator as shown below

set3 = {'a', 'b', 'c'}  
for i in range(1,5):  
 set3.add(i)  
print(set3)

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

* Add() - adds an individual element to the set but it cannot work with an iterable

a.add(1) works but a.add([1, 2]) wont work

* update() – we can pass multiple iterables to it and it will iterate all iterables and will include the individual elements in the set.

a.update([1]) works

*syntax: set*.update(*set*)

Q50. What is clear() in sets?

* Python Set clear() method removes all elements from the set

Q51. What is frozen set?

* Python frozenset() Method creates an immutable Set object from an iterable. It is a built-in Python function. As it is a set object therefore we cannot have duplicate values in the frozenset.

**Syntax :** frozenset(iterable\_object\_name)  
**Parameter :** iterable\_object\_name

Q52. How is frozen set different from set?

* A set is a mutable object while frozenset provides an immutable

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

* The Python set union() method returns a new set with distinct elements from all the sets
* set1 = {1,2,3,4,5}  
  set2 = {3,4,5,6,7}  
  unionset = set1.union(set2)  
  print(unionset)

{1, 2, 3, 4, 5, 6, 7}

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

* The intersection() method returns a new set with elements that are common to all sets.
* set1 = {1,2,3,4,5}  
  set2 = {3,4,5,6,7}  
  intersectionset = set1.intersection(set2)

{3, 4, 5}

Q55. What is dictionary ibn Python?

* **Dictionary in Python** is a collection of keys values, used to store data values like a map, which, unlike other data types which hold only a single value as an element.
* Dictionary holds **key:value** pair

For eg: Dict = {Name: 'Shiju', Place: 'USA', Profession: 'Software Engineer'}

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

| # | Array | Dictionary |
| --- | --- | --- |
| 1 | Stores just a set of objects | Represents the relationship between pair of objects |
| 2 | Lookup time is more in the  case of array O(N)  where N is the size of the array | Lookup time is less compared to an array.  Generally, it is O(1) |
| 3 | Elements are stored at contagious memory locations. | Elements may or may not be stored at a contagious memory location. |
| 4 | Items are unordered, changeable, and do allow duplicates | Items are ordered, changeable, and do not allow duplicates |
| 5 | Items are not represented as key: value pair | Items are represented as key: value pair |
| 6 | The values in the array are of the same data type | The values in dictionary items can be of any data type |
| 7 | Values can be accessed randomly without the need for any key | To access a value the key is required |

Q57. How can we delare a dictionary in Python?

* A Dictionary in python is declared by enclosing a comma-separated list of key-value pairs using curly braces({}). Python Dictionary is classified into two elements: Keys and Values.
* Keys will be a single element
* Values can be a list or list within a list, numbers, etc.

Q58. What will the output of the following?

```

var = {}

print(type(var))

```

* <class 'dict'>

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

* Addition of elements can be done in multiple ways. One value at a time can be added to a Dictionary by defining value along with the key e.g. Dict[Key] = ‘Value’. Updating an existing value in a Dictionary can be done by using the built-in **update()** method. Nested key values can also be added to an existing Dictionary.
* **Note-** While adding a value, if the key-value already exists, the value gets updated otherwise a new Key with the value is added to the Dictionary.

Eg:

# Creating an empty Dictionary  
Dict = {}  
print("Empty Dictionary: ")  
print(Dict)

Empty Dictionary:

{}

# Adding elements one at a time  
Dict[0] = 'India'  
Dict[2] = 'For'  
Dict[3] = 1  
print("\nDictionary after adding 3 elements: ")  
print(Dict)

Dictionary after adding 3 elements:

{0: 'India', 2: 'For', 3: 1}

# Adding set of values  
# to a single Key  
Dict['Value\_set'] = 2, 3, 4  
print("\nDictionary after adding 3 elements: ")  
print(Dict)

Dictionary after adding 3 elements:

{0: 'India', 2: 'For', 3: 1, 'Value\_set': (2, 3, 4)}

# Updating existing Key's Value  
Dict[2] = 'Welcome'  
print("\nUpdated key value: ")  
print(Dict)

Updated key value:

{0: 'India', 2: 'Welcome', 3: 1, 'Value\_set': (2, 3, 4)}

# Adding Nested Key value to Dictionary  
Dict[5] = {'Nested': {'1': 'Life', '2': 'Enjoy'}}  
print("\nAdding a Nested Key: ")  
print(Dict)

Adding a Nested Key:

{0: 'India', 2: 'Welcome', 3: 1, 'Value\_set': (2, 3, 4), 5: {'Nested': {'1': 'Life', '2': 'Enjoy'}}}

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

#Create a dictionary and access all the values in that dictionary  
dict1 ={'Key1':'Value1', 'key2':'Value2', 'key3':'value3'}  
print(dict1)

#Create a dictionary and access all the values in that dictionary  
dict1 ={'uid':'01242534647', 'name':'Shiju Pananjikal', 'country':'USA'}  
print(dict1)

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

#Create a dictionary and access all the values in that dictionary  
dict1 ={'uid':'01242534647', 'name':'Shiju Pananjikal', 'country':'USA'}  
print(dict1)  
  
#nested dictionary and access all the element in the inner dictionary  
dict1['Nested'] = {'Skills':'Python', 'relocation':'Yes', 'Immigration\_status':'NA'}  
print(dict1)

{'uid': '01242534647', 'name': 'Shiju Pananjikal', 'country': 'USA'}

{'uid': '01242534647', 'name': 'Shiju Pananjikal', 'country': 'USA', 'Nested': {'Skills': 'Python', 'relocation': 'Yes', 'Immigration\_status': 'NA'}}

#nested dictionary and access all the element in the inner dictionary  
dict1 ={'uid':'01242534647', 'name':'Shiju Pananjikal', 'country':'USA','Nested':{'Skills':'Python', 'relocation':'Yes', 'Immigration\_status':'NA'}}  
  
  
print(dict1)  
  
print (dict1['uid'])  
print (dict1['name'])  
print (dict1['country'])  
print (dict1['Nested']['Skills'])  
print (dict1['Nested']['relocation'])  
print (dict1['Nested']['Immigration\_status'])

{'uid': '01242534647', 'name': 'Shiju Pananjikal', 'country': 'USA', 'Nested': {'Skills': 'Python', 'relocation': 'Yes', 'Immigration\_status': 'NA'}}

01242534647

Shiju Pananjikal

USA

Python

Yes

NA

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

**Python Dictionary get() Method** return the value for the given key if present in the dictionary. If not, then it will return None (if get() is used with only one argument).

dict1 ={'uid':'01242534647', 'name':'Shiju Pananjikal', 'country':'USA','Nested':{'Skills':'Python', 'relocation':'Yes', 'Immigration\_status':'NA'}}  
  
print(dict1.get('country'))  
nested\_get = dict1.get('Nested',{}).get('Skills')  
print(nested\_get)

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

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

dict1 ={'uid':'01242534647', 'name':'Shiju Pananjikal', 'country':'USA','Nested':{'Skills':'Python', 'relocation':'Yes', 'Immigration\_status':'NA'}}  
items = dict1.items()  
print(items)

dict\_items([('uid', '01242534647'), ('name', 'Shiju Pananjikal'), ('country', 'USA'), ('Nested', {'Skills': 'Python', 'relocation': 'Yes', 'Immigration\_status': 'NA'})])

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

* Python dictionary **pop()**method removes and returns the specified element from the dictionary.
* dict1 ={'uid':'01242534647', 'name':'Shiju Pananjikal', 'country':'USA','Nested':{'Skills':'Python', 'relocation':'Yes', 'Immigration\_status':'NA'}}  
  pop\_object = dict1.pop('country')  
  print(dict1)

{'uid': '01242534647', 'name': 'Shiju Pananjikal', 'Nested': {'Skills': 'Python', 'relocation': 'Yes', 'Immigration\_status': 'NA'}}

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

* Python dictionary popitem() method removes the last inserted key-value pair from the dictionary and returns it as a tuple.
* dict1 ={'uid':'01242534647', 'name':'Shiju Pananjikal', 'country':'USA','Nested':{'Skills':'Python', 'relocation':'Yes', 'Immigration\_status':'NA'}}  
  popitem = dict1.popitem()  
  print('The key value pair returned is:', popitem)  
  print('The dict1 after popitem',dict1)

The key value pair returned is: ('Nested', {'Skills': 'Python', 'relocation': 'Yes', 'Immigration\_status': 'NA'})

The dict1 after popitem {'uid': '01242534647', 'name': 'Shiju Pananjikal', 'country': 'USA'}

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

* The **keys()** method in [Python Dictionary](https://www.geeksforgeeks.org/python-dictionary/), returns a view object that displays a list of all the keys in the dictionary in order of insertion using [Python](https://www.geeksforgeeks.org/python-programming-language/).
* dict1 ={'uid':'01242534647', 'name':'Shiju Pananjikal', 'country':'USA','Nested':{'Skills':'Python', 'relocation':'Yes', 'Immigration\_status':'NA'}}  
    
  keys = dict1.keys()  
  print(keys)

dict\_keys(['uid', 'name', 'country', 'Nested'])

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

* The view object contains the values of the dictionary, as a list.
* dict1 ={'uid':'01242534647', 'name':'Shiju Pananjikal', 'country':'USA','Nested':{'Skills':'Python', 'relocation':'Yes', 'Immigration\_status':'NA'}}  
  values\_dict = dict1.values()  
  print(values\_dict)

dict\_values(['01242534647', 'Shiju Pananjikal', 'USA', {'Skills': 'Python', 'relocation': 'Yes', 'Immigration\_status': 'NA'}])

Q68. What are loops in Python?

* A loop is an instruction that repeats multiple times as long as some condition is met.

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

* There are two types of loops, While loop and For loop

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

While loop:

* In python, a [while loop](https://www.geeksforgeeks.org/python-while-loop/) is used to execute a block of statements repeatedly until a given condition is satisfied. And when the condition becomes false, the line immediately after the loop in the program is executed.

For [loops](https://www.geeksforgeeks.org/python-for-loops/):

* For loops are used for sequential traversal. For example: traversing a list or string or array etc.

Q71. What is the use of continue statement?

* It returns the control to the beginning of the loop.
* # Prints all letters except 'a' and 'n'  
  for letter in 'pananajikal':  
   if letter == 'a' or letter == 'n':  
   continue  
   print('Current Letter :', letter)

Current Letter : p

Current Letter : j

Current Letter : i

Current Letter : k

Current Letter : l

Q72. What is the use of break statement?

* It brings control out of the loop
* # break the loop as soon as it sees 'j' or 'k'  
  for letter in 'pananajikal':  
   if letter == 'j' or letter == 'k':  
   break  
  print('Current Letter :', letter)

Current Letter : j

Q73. What is the use of pass statement?

* We use pass statement to write empty loops. Pass is also used for empty control statements, functions and classes.
* the pass statement is a null statement which can be used as a placeholder for future code.
* Suppose we have a loop or a function that is not implemented yet, but we want to implement it in the future. In such cases, we can use the pass statement.

n = 10  
# use pass inside if statement  
if n > 10:  
 pass  
print('Hello')

Hello

n = 10  
if n > 10:  
 # write code later  
print('Hello')

print('Hello')

^

IndentationError: expected an indented block

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

* The Python**range() function**returns a sequence of numbers, in a given range. The most common use of it is to iterate sequence on a sequence of numbers using [Python](https://www.geeksforgeeks.org/python-programming-language/) loops.
* **Syntax:** range(start, stop, step)

Q75. How can you loop over a dictionary?

* We can loop through a dictionary **by using a for loop**. When looping through a dictionary, the return value are the keys of the dictionary.

### Coding problems

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

#Write a Python program to find the factorial of a given number.  
  
number = int(input('Enter the number:\n'))  
fact = 1  
for i in range(1,number+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

# Write a Python program to calculate the simple interest. Formula to calculate simple interest is SI = (P\*R\*T)/100  
  
def simple\_interest(P,R,T):  
 SI = (P \* R\* T)/100  
 return SI  
  
principal\_amt = 1000  
Interest\_amt = 5 # in percentage  
Time = 5 # in years  
  
print('simple interest:', simple\_interest(principal\_amt,Interest\_amt,Time))

simple interest: 250.0

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

# Write a Python program to calculate the compound interest. Formula of compound interest is A = P(1+ R/100)^t.  
  
def compound\_interest(P,R,t):  
 A = P\*(1+ R/100)\*\*t  
 comp\_int = A - P  
 return round(comp\_int)  
  
principal\_amt = 1000  
Interest\_amt = 5 # in percentage  
Time = 5 # in years  
  
print('compound interest:', compound\_interest(principal\_amt,Interest\_amt,Time))

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

# Write a Python program to check if a number is prime or not.  
  
num = int(input('provide the number:'))  
if num > 1:  
  
 for i in range (2, int(num/2)+1 ):  
 if (num % i) == 0:  
 print(num, 'is not prime number')  
 break  
 else:  
 print(num, 'is prime number')  
else:  
 print(num, 'is not a prime number')

Q80. Write a Python program to check Armstrong Number.

# Write a Python program to check Armstrong Number.  
# Armstrong Numbers are the numbers having the sum of digits raised to power no. of digits is equal to a given number.  
#eg: 153 1\*\*3+ 5\*\*3 +3\*\*3 = 1+ 125+ 27 = 153, hence 153 is Armstrong number  
  
num = int(input("Enter the number:"))  
  
arm\_num = (list(map(int,str(num)))) # convert the number to list, 153 to [1,5,3]  
  
sum\_num = 0  
for i in arm\_num:  
 sum\_num = sum\_num + i\*\*(len(str(num))) # multiply each digit with length of the given num times and sum  
if sum\_num == num:   
 print(num, "is Armstrong number")  
else:  
 print(num, "is not Armstrong number")

153 is Armstrong number

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

#Write a Python program to find the n-th Fibonacci Number.  
  
num = int(input('Enther the number:'))  
fib\_num = [0,1]  
def fiboacci\_num (num):  
  
 fib\_num = [0, 1]  
 for i in range(2,num+1):  
 fib\_num.append(fib\_num[i-1]+fib\_num[i-2])  
 return fib\_num[num]  
  
n= fiboacci\_num(num)  
print(num,'th fibonacci number is', n)

Enther the number: 9

9 th fibonacci number is 34

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

# Write a Python program to interchange the first and last element in a list.  
  
def swap\_list():  
  
 size = len(list)  
 temp = list[0]  
 list[0] = list[size-1]  
 list[size-1] = temp  
 return(list)  
  
list = [1, 2, 'Shiju', 4, '5', 6, 'Pananjikal']  
  
list\_new = swap\_list()  
print(list\_new)

['Pananjikal', 2, 'Shiju', 4, '5', 6, 1]

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

#Write a Python program to swap two elements in a list.  
  
def swap\_elements(list,pos1, pos2):  
 list = ['a',1,3,5,'b','c']  
 temp = list[pos1]  
 list[pos1] = list[pos2]  
 list[pos2] = temp  
 return list  
pos1,pos4 = 2,5  
  
print(swap\_elements(list,pos1-1,pos4-1))

['a', 'b', 3, 5, 1, 'c']

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

# Write a Python program to find N largest element from a list.  
  
list = [3,6,1,8,10,22,5]  
N = 3  
  
def largest\_elements(list, N):  
  
 new\_list = []  
 for j in range(0,N):  
 largest\_element = 0  
  
 for i in range(len(list)):  
 if list[i] > largest\_element:  
 largest\_element = list[i]  
  
 new\_list.append(largest\_element)  
 list.remove(largest\_element)  
 return new\_list  
  
print(largest\_elements(list,N))

[22, 10, 8]

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

# Write a Python program to find cumulative sum of a list.  
  
# Input: 1, 2, 3, 4, 5 6, 7 ,8, 9, 10  
# Output: 1, 3, 6, 10, 15, 21, 28, 36, 45, 55  
list = [1, 2, 3, 4, 5, 6, 7 ,8, 9, 10]  
def cumulative\_sum ():  
 sum = 0  
 new\_list =[]  
 for i in range (0,len(list)):  
 sum = sum + list[i]  
 new\_list.append(sum)  
 return new\_list  
  
print(list,'\n',cumulative\_sum ())

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

[1, 3, 6, 10, 15, 21, 28, 36, 45, 55]

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

#Write a Python program to check if a string is palindrome or not.  
  
string = 'malayalam'  
  
def palindrom\_check():  
  
 return string == string[::-1]  
  
reverse\_string = palindrom\_check()  
  
if reverse\_string:  
 print('The given string is palindrome')  
else:  
 print('The given string is not palindrome')

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

#Write a Python program to remove i'th element from a string.  
ithelement = int(input("Enter the position of an element in the string:"))  
  
def remove\_ithelement():  
 string = 'pythonprogram'  
 for i in range(len(string)+1):  
 if i == ithelement:  
 string\_new = string[0:i] + string[i+1:]  
 return string\_new  
  
print(remove\_ithelement())

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

#Write a Python program to check if a substring is present in a given string.  
string = 'Python program to check if a substring is present in a given string'  
sub\_strg ='sub'  
  
if sub\_strg in string:  
 print (sub\_strg, 'is presnet in the given string', string)

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

#Write a Python program to find words which are greater than given length k.  
string = 'Python program to find words which are greater than given length k'  
length = 5  
  
string\_new = string.split(" ")  
  
for i in range (len(string\_new)):  
 if len(string\_new[i]) > 5:  
 print(string\_new[i])

Python

program

greater

length

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

#Python program to extract unique dictionary values.  
dictionary = [{'Name': 'Shiju', 'Year': 2023}, {'Dept': 'IT', 'UID': 2023}, {'Skills': 'Python'}]  
  
dict\_values = []  
  
for i in dictionary:  
 dict\_values.extend(list(i.values()))  
unique\_set = set(dict\_values)  
  
print(unique\_set)

{'Shiju', 'Python', 'IT', 2023}

Q91. Write a Python program to merge two dictionary.

#Write a Python program to merge two dictionary  
  
def merge\_dictionary (dict1, dict2):  
 merge\_dict = {\*\*dict1, \*\*dict2}  
 return merge\_dict  
  
dict1 = {'Name': 'Shiju', 'Dept': 'IT'}  
dict2 = {'Skills' : 'Python', 'UID' : 200120120}  
print (merge\_dictionary(dict1,dict2))

{'Name': 'Shiju', 'Dept': 'IT', 'Skills': 'Python', 'UID': 200120120}

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}

```

#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}  
Dict = dict(Input)  
print('Input :', Input)  
print ('Output :', Dict)

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

Output : {'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)]

```

# 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)]  
  
list = [9, 5, 6]  
list\_new = [(i,i\*\*3) for i in list]  
print(list\_new)

[(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)]

```

#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)  
  
tuple\_result1 = [(i,j) for i in test\_tuple1 for j in test\_tuple2]  
tuple\_result2 = [(i,j) for i in test\_tuple2 for j in test\_tuple1]  
tuple\_result = tuple\_result1 + tuple\_result2  
  
print(tuple\_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)]

```

#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)]  
  
list = [('for', 24), ('Geeks', 8), ('Geeks', 30)]  
  
for i in range (0,len(list)):  
 for j in range (0,len(list)-i-1 ):  
 if (list[j][1] > list[j+1][1]):  
 temp = list[j]  
 list[j] = list[j+1]  
 list[j+1] = temp  
  
  
print(list)

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

Q96. Write a python program to print below pattern.

```

\*

\* \*

\* \* \*

\* \* \* \*

\* \* \* \* \*

```

# Write a python program to print below pattern.  
#  
# \*  
# \* \*  
# \* \* \*  
# \* \* \* \*  
# \* \* \* \* \*  
  
def pyramid\_pattern(n):  
 for i in range(0,n):  
 for j in range (0,i+1):  
 print("\* ",end=" ")  
 print("\r")  
n = 5  
  
pyramid\_pattern(n)

Q97. Write a python program to print below pattern.

```

\*

\*\*

\*\*\*

\*\*\*\*

\*\*\*\*\*

```

Q98. Write a python program to print below pattern.

```

\*

\* \*

\* \* \*

\* \* \* \*

\* \* \* \* \*

```

# Write a python program to print below pattern.  
# \*  
# \* \*  
# \* \* \*  
# \* \* \* \*  
# \* \* \* \* \*  
  
  
def trinagle\_pattern1(n):  
 k = n-1 # no of spaces  
 for i in range(0,n): #outerloop for spaces  
 for j in range (0,k): #innerloop for spaces  
 print(end=" ")  
 k = k-1 #decrementing k after each loop  
  
 for j in range(0, i+1): #inner loop to handle number of columns  
 print("\* ", end = "")  
 print("\r")  
n = 5  
  
trinagle\_pattern1(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

``` # Write a python program to print below pattern.  
# ```  
# 1  
# 1 2  
# 1 2 3  
# 1 2 3 4  
# 1 2 3 4 5  
  
  
def number\_pattern(n):  
  
 for i in range(n): # outer loop  
 p = 1 # initialize with the first number required in the row  
 for j in range (i+1): # inner loop  
 print(p, end=' ') # use the variable p to print the numbers  
 p+=1 # increment p to get it to the next row  
 print()  
  
n=5  
number\_pattern(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

```

# Write a python program to print below pattern.  
#  
# A  
# B B  
# C C C  
# D D D D  
# E E E E E  
  
def character\_pattern(n):  
  
 p = 65 # ascii code of characters A-Z = 65-90  
 for i in range(n): # outer loop  
 for j in range (i+1): # inner loop  
 print(chr(p), end=' ') # use the variable p to print the Alphabets of the corresponding ascii #  
 p+=1 # increment the ascii number of the alphabets  
 print()  
  
n=5  
character\_pattern(n)

A

B B

C C C

D D D D

E E E E E