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

A1. Python is a general purpose programming language as it can be used to build software in a wide variety of application domains such as data science, web development, software development, machine learning etc.   
Python is a high-level programming language as it is easily understood by humans but computers can not directly understand it. It has to be converted to machine code or low level language for it to be understood by a computer.

**Q2. Why is Python called a dynamically typed language?**

A2. Python is called a dynamically typed language because in Python an interpreter assigns variables a type at runtime based on the variable's value at the time & the type of a variable is allowed to change over its lifetime.

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

A3.

Pros -

* Free and open source
* Beginner friendly - It is easy to learn, understand and code as it does not include too many technicalities.
* Extensive Libraries
* Portable - Write once run anywhere.
* Highly scalable
* Dynamically typed - As the programmer, you don’t have to worry about the declaration of variables and the data types.

Cons -

* Poor Memory Efficiency
* Slow speed - It is slow primarily due to its dynamic nature and versatility.
* Weak in Mobile Computing
* Runtime Errors - A variable holding a string may contain an integer later, and this can lead to runtime errors.

**Q4. In what all domains can we use Python?**

A4. Python can be used in a wide variety of domains. Some of them are -

* Data science, Data analytics, data visualization.
* Web development.
* Game development.
* Machine learning / Artificial intelligence.
* Mobile app development.
* Desktop GUI.

**Q5. What are variables and how can we declare them?**

A5. Variable is a name given to a memory location.

Python has no command for declaring a variable. A variable is created the moment you first assign a value to it.

Eg. a = 5

b = “Tom”

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

A6. Input() function is used to take user input in python.

Syntax - input(prompt)

Eg. name = input("Enter name = ")

print("User name = ",name)

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

A7. String

**Q8. What is type casting?**

A8. Type casting is a method to convert datatype of a variable.

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

A9. Yes, split () can be used to take multiple inputs from a user using single input ().

Eg. x, y = input("Enter the number of boys and girls: ").split()

print("Number of boys: ", x)

print("Number of girls: ", y)

**Q10. What are keywords?**

A10. Python keywords are special reserved words that have specific meanings and purposes and can't be used for anything but those specific purposes.

Eg. break, continue, else, elif, for, if, in etc.

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

A11. No, keywords can’t be used as a variable in python because keywords have specific meaning and purpose in python.

Eg. while is a keyword in python used to create a loop.

while = 5

print (while)

This code would not work and throw a syntax error because python understands that while has a specific function and specific syntax and therefore it will try to execute it for looping and not as a variable holding a value.

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

A12. Indentation is simply the spaces at the beginning of a code line. It is used for readability, but in Python, the indentation is an essential and mandatory.

Python uses indentation to indicate a block of code. Python treats the statements with the same indentation level (statements with an equal number of whitespaces before them) as a single code block.

Eg.

if x==y:

print("Yes, X is Equals to Y !!")

else:

print("No, X is not Equals to Y !!")

The print statement under the if statement is if code block similarly the print statement under else is the else code block.

**Q13. How can we throw some output in Python?**

A13. print () can be used to throw output in Python.

Eg.

name = "Tom"

print(name)

Output - Tom

**Q14. What are operators in Python?**

A14. Operators are used to perform operations on variables and values.

There are multiple operators in python -

* Arithmetic operators
* Assignment operators
* Comparison operators
* Logical operators
* Identity operators
* Membership operators
* Bitwise operators

Eg.

x = 5

y = 3

print("Addition of x + y = ", x+y)

Output - Addition of x + y = 8

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

A15. / is for float division &

// is for integer division.

Eg.

x = 5

y = 3

print("Float Division of x / y = ", x/y)

print("Integer Division of x // y = ", x//y)

Output - Float Division of x / y = 1.6666666666666667

Integer Division of x // y = 1

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

**```**

**iNeuroniNeuroniNeuroniNeuron**

A16.

str1 = "iNeuron"\*4

print("Multiply str = ", str1)

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

A17.

num = int(input("Enter any number ="))

if num % 2 == 0 :

print("The provided number is even")

else :

print("The provided number is odd")

**Q18. What are boolean operators?**

A18. Boolean Operators are those that result in the Boolean values of True and False.

There are two types of operators in Python that return boolean values, i.e., Logical operators and Comparison operators. Since both these operators have return types as boolean, they are also termed Boolean operators.

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

**```**

**1 or 0**

**0 and 0**

**True and False and True**

**1 or 0 or 0**

A19.

| 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?**

A20. These statements guide the program while making decisions based on the conditions encountered by the program.

Using conditional statements, the program can be routed to different steps based on check of conditional statements.

Python has 3 key Conditional Statements:

If statement

If-else statement

Nested if-else statement

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

A21. If, else and elif keywords are used as conditional statements to determine whether a block of code will be executed or not.

Else - Is used to execute else code block when the “if” conditions fail.

Elif - Is used in nested if else statement i.e. to make multiple conditional statements.

**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".**

A22.

Age = int(input("Enter 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]**

**```**

A23.

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

even = []

for num in numbers:

if num % 2 == 0:

even.append(num)

print(even)

total = 0

for el in range(0, len(even)):

total = total + even[el]

print(total)

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

A24.

x, y, z = [int(x) for x in input("Enter 3 numbers: ").split()]

if x>y and x>z:

print(x, "is the largest among the entered numbers")

elif y > z and y > x:

print(y, "is the largest among the entered numbers")

else:

print(z, "is the largest among the entered numbers")

**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]**

A25.

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

for x in numbers:

if x > 500:

break

elif x > 150:

continue

elif x % 5 == 0:

print(x)

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

A26. String is a text type data type. It is a sequence of characters where characters can be alphabets, words or other characters.

String can be declared by surrounding the characters with single or double quotes and assigning it to a variable.

Eg.

str1 = "hello"

str2 = 'bye'

Multiline strings can be declared using triple quotes.

Eg.

""" This is

a multiline string """

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

A27. String can be accessed using indexes similar to lists. Index starts with 0 i.e. the first character has 0 index.

Eg.

str1 = 'hello'

#print 1st element

print(str1[0])

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

**string = "Big Data iNeuron"**

**desired\_output = "iNeuron"**

A28.

string = "Big Data iNeuron"

start\_index = string.find("iNeuron")

len\_word = len("iNeuron")

print(string[start\_index:start\_index+len\_word])

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

**string = "Big Data iNeuron"**

**desired\_output = "norueNi"**

A29.

string = "Big Data iNeuron"

start\_index = string.find("iNeuron")

len\_word = len("iNeuron")

print(string[start\_index+len\_word:start\_index-1:-1])

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

A30.

string = "Big Data iNeuron"

print(string[len(string): :-1])

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

A31.

Entire string can be deleted using the del command.

**Q32. What is escape sequence?**

A32.

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 that may be difficult or impossible to express directly, like newline (\n), tab (\t), and so on.

Escape sequences allow you to insert special characters in strings. Put a backslash (\) before the character you want to escape.

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

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

A33.

print("iNeuron's Big Data Course")

OR

print('iNeuron\'s Big Data Course') #using escape sequence

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

A34.

Lists are one of 4 built-in data types in Python used to store collections of data.

List items are indexed, ordered, changeable, and allow duplicate values.

1st element has 0 index.

List items can be of any datatype.

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

A35.

Lists can be created using square brackets.

L1 = [1, 2, 'tom', 1.5, 'jerry']

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

A36. List items can be accessed using index.

Eg.

L1 = [1, 2, 'tom', 1.5, 'jerry']

print(L1[1])

Multiple elements can be accessed using range of indexes or list slicing.

Syntax -   
List\_name[start\_index:end\_index:step]

Eg.

L1 = [1, 2, 'tom', 1.5, 'jerry']

print(L1[1:4:2])

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

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

A37.

print(lst[4][2])

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

A38.

x = [x for x in input("Enter list elements separated by whitespaces: ").split()]

print(len(x))

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

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

A39.

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

lst.insert(2, "Big")

print(lst)

#Output - ['Welcome', 'to', 'Big', 'Data', 'course'] Big is 3rd element and its index is 2

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

A40. Tuple is also one of the 4 built-in data types in Python used to store collections of data.   
Tuple items are ordered, indexed, unchangeable, and allow duplicate values whereas list items are changeable in addition to being ordered and allowing duplicate values.

Tuples are written with round brackets.

Eg. tuple1 = ("tom", "jerry", "ben")

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

A41. Tuples can be created using round brackets. Tuple can have any datatype as elements.

Note - To create a tuple with only one item, add a comma after the item, otherwise Python will not recognize it as a tuple.

Eg. tuple1 = ("tom", "jerry", "ben")

tuple2 = ("jack",)

print(type(tuple1))

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

A42. Tuple is unchangeable and immutable. Therefore a new element cannot be added to an existing tuple.

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

A43.

tuple1 = ("tom", "jerry", "ben")

t2 = ("gwen",)

print(tuple1 + t2)

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

A44.

x = input("Enter tuple elements separated ").split()

y = tuple(x)

print(y)

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

A45. Set is one of 4 built-in data types in Python used to store collections of data.   
Set items are unordered, unchangeable(existing items cannot be changed but new items or remove items), and do not allow duplicate values.

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

A46.

set1 = {"tom", 7, "kilo"}

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

A47.

set1 = set()

set1.add("iNeuron")

print(set1)

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

A48. Only one item can be added at once using add().

set1 = set()

set1.add("iNeuron")

set1.add("Batch 2")

print(set1)

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

A49. update() is add items from another set into current set whereas add() is used to add individual element directly into current set.

Eg.

set1 = set()

set1.add("iNeuron")

set1.add("Batch 2")

set2 = {"Welcome",}

set1.update(set2)

print(set1)

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

A50.

The clear() method empties the set.

set1 = set()

set1.add("iNeuron")

set1.add("Batch 2")

set2 = {"Welcome",}

set1.update(set2)

set1.clear()

print(set1)

**Q51. What is frozen set?**

A51. Python frozenset() Method creates an immutable Set object from an iterable.

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

A52.

In a set, items can be added or removed but a frozen set is immutable.

Frozen set can be used as a key in dictionary.

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

A53. Union() can be used to join two sets. It returns a new set with all items from both sets.

set1 = {"iNeuron"}

set2 = {"Batch 2"}

set3 = set1.union(set2)

print(set3)

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

A54. The intersection() method will return a new set, that only contains the items that are present in both sets.

set1 = {"iNeuron", "tom"}

set2 = {"Batch 2", "tom"}

set3 = set1.intersection(set2)

print(set3)

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

A55. Dictionary is one of the built-in datatype in python. It is used to store data values in key:value pairs.

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

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

A56. Dictionary stores data in key value pairs whereas other data structures like list, tuple etc. store single elements.

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

A57. Dictionary can be declared using curly brackets.

dict1 = {

"name": "Tom",

"age": 22

}

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

var = {}

print(type(var))

A58.

<class 'dict'>

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

A59.

dict1 = {}

dict1["name"] = "jerry"

dict1["age"] = 24

print(dict1)

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

A60.

dict1 = {}

dict1["name"] = "jerry"

dict1["age"] = 24

print(dict1)

print(type(dict1))

for k, v in dict1.items():

print("key is {}, value is {}".format(k, v))

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

A61.

dict1 = {

"brand" : "VW",

"model" : "polo",

"features" : {"seat":5, "engine":"1200cc", "fuel":"petrol"}

}

print(dict1["features"])

print("elements of inner dictionary: ")

for k, v in dict1["features"].items():

print("key is {}, value is {} ".format(k,v))

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

A62. The get() method returns the value of the item with the specified key.

Syntax -

dictionary.get(key, value)

Value input is optional if you want get() to return some value if the key is not present.

dict1 = {

"brand" : "VW",

"model" : "polo",

"features" : {"seat":5, "engine":"1200cc", "fuel":"petrol"}

}

x = dict1.get("model")

print(x)

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

A63.

The items() method returns a view object. The view object contains the key-value pairs of the dictionary, as tuples in a list.

dict1 = {

"brand" : "VW",

"model" : "polo",

"features" : {"seat":5, "engine":"1200cc", "fuel":"petrol"}

}

x = dict1.items()

print(x)

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

A64. The pop() method removes the specified item from the dictionary. The value of the removed item is the return value of the pop() method.

dict1 = {

"brand" : "VW",

"model" : "polo",

"features" : {"seat":5, "engine":"1200cc", "fuel":"petrol"}

}

x = dict1.pop("features")

print(dict1)

print(x)

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

A65. The popitem() method removes the item that was last inserted into the dictionary.

dict1 = {

"brand" : "VW",

"model" : "polo",

"features" : {"seat":5, "engine":"1200cc", "fuel":"petrol"}

}

x = dict1.popitem()

print(dict1)

print(x)

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

A66. The keys() method returns a view object. The view object contains the keys of the dictionary.

dict1 = {

"brand" : "VW",

"model" : "polo",

"features" : {"seat":5, "engine":"1200cc", "fuel":"petrol"}

}

x = dict1.keys()

print(x)

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

A67. The values() method returns a view object. The view object contains the values of the dictionary.

dict1 = {

"brand" : "VW",

"model" : "polo",

"features" : {"seat":5, "engine":"1200cc", "fuel":"petrol"}

}

x = dict1.values()

print(x)

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

A68. Loops are used to execute a block of statements repeatedly.

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

A69. 2. For and while loop.

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

A70. When the number of iterations is known for loop is used whereas While is used when number of iterations are not known.

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

A71. Continue statement is used in a loop to skip the block of code that is written after the continue statement and to move back the control back to loop.

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

A72. Break statement is used in a loop to stop the loop i.e. to move the control out of the loop.

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

A73. The pass statement is used as a placeholder for future code.

When the pass statement is executed, nothing happens, but you avoid getting an error when empty code is not allowed.

Empty code is not allowed in loops, function definitions, class definitions, or in if statements.

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

A74. The range() function returns a sequence of numbers, starting from 0 by default, and increments by 1 (by default), and stops before a specified number.

Syntax -

range(start, stop, step)  
It is typically used to loop over a sequence of numbers.

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

A75.

dict1 = {

"brand" : "VW",

"model" : "polo",

"features" : {"seat":5, "engine":"1200cc", "fuel":"petrol"}

}

for keys in dict1:

print(keys)

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

A76.

n = int(input("Enter any positive integer: "))

fact = 1

for x in range(n, 0, -1):

fact = fact \* x

print("Factorial of {} = {}".format(n, fact))

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

A77.

p, r, t = [float(x) for x in input("Enter the principal amount, rate of interest and time period separated by whitespaces: ").split()]

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

print("Simple interest is = ", SI)

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

A78.

def CI(p, r, t):

A = p\*(1+r/100)\*\*t

I = A - p

print("Compound interest = ", I)

return I

CI(1000, 9.5, 2)

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

A79.

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

count = 0

for x in range(1, num+1):

if num%x == 0:

count = count + 1

if count == 2:

print("Prime number")

else:

print("Not a prime number")

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

A80.

x = input("Enter a number to check if its an Armstrong number: ") #taking str input

L1 = list((x)) #ingesting the string in a list as individual characters

L2 = list(map(int, L1)) #converting list elements to int

add = 0

for i in L2:

add = add + i\*\*len(L2)

print("Sum of digits raised to power of number of digits =", add)

if add == int(x):

print("{} is an Armstrong number".format(x))

else:

print("{} is not an Armstrong number".format(x))

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

A81.

n = int(input("Enter the nth value for Fibonacci sequence: "))

L1 = [0, 1]

i = 0

while len(L1) < n:

x = L1[i] + L1[i+1]

L1.append(x)

i = i+1

print(L1)

print(L1[n-1])

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

A82.

def swap(a):

temp = a[0]

a[0] = a[len(a)-1]

a[len(a)-1] = temp

return a

L1 = [1, 2, 3, 4, 5]

print(swap(L1))

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

A83.

def swap\_items(list, a, b):

temp = list[a]

list[a] = list[b]

list[b] = temp

return list

L1 = [1, 2, 3, 4, 5, 6]

a, b = 2, 5

swap\_items(L1, a, b)

print("list after item swapping:", L1)

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

A84.

def n\_max(thelist, n):

max\_list = []

for i in range(0, n):

max = 0

for j in range(0, len(thelist)):

if thelist[j] > max:

max = thelist[j]

thelist.remove(max)

max\_list.append(max)

return max\_list

L1 = [15, 20, 5, 67, 90, 3, 54]

n = 3

print(n\_max(L1, n))

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

A85.

def cumulative\_sum(L):

new\_list = []

total = 0

for i in L:

total += i

new\_ele = total

new\_list.append(total)

return new\_list

L1 = [1, 2, 3, 4, 5]

print(cumulative\_sum(L1))

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

A86.

def palindrome\_str(stringin):

rev = stringin[-1: :-1]

if rev == stringin:

print("{} is a palindrome".format(stringin))

else:

print("{} is not a palindrome".format(stringin))

str1 = "tenet"

palindrome\_str(str1)

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

A87.

def remove\_str\_ele(str1, i):

L\_str = list(str1)

L\_str.pop(i-1)

result = ''.join(L\_str)

print(result)

str1 = "tenet"

i = 4

remove\_str\_ele(str1, i)

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

A88.

def check\_str(str1, substr):

if substr in str1:

print("present")

else:

print("not present")

str1 = 'The Weeknd'

substr = 'Weeknd'

check\_str(str1, substr)

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

A89.

def check\_len(str1, k):

L\_str = str1.split()

for i in L\_str:

if len(i) > k:

print(i)

str1 = 'Welcome to the batch'

k =4

check\_len(str1, k)

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

A90.

dict2 = {'name1':'tom', 'name2':'tom', 'name3':'jerry', 'age1':22, 'age2':24, 'age3':22}

set\_dict2 = set(dict2.values())

print(set\_dict2)

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

A91.

def Merge(dict1, dict2):

merge\_dict = dict2.update(dict1)

return(merge\_dict)

dict1 = {'name1':'Tom', 'age1':24}

dict2 = {'name2':'Jerry', 'age2':25}

print(Merge(dict1, dict2))

print(dict2)

**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}**

A92.

list\_tuple = [('Sachin', 10), ('MSD', 7), ('Kohli', 18), ('Rohit', 45)]

dict\_tuple = dict(list\_tuple)

print(dict\_tuple)

**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)]**

A93.

input\_list = [9, 5, 6]

dict\_list = {}

for i in input\_list:

dict\_list[i] = i\*\*3

print(list(tuple(dict\_list.items())))

**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)]**

A94.

test\_t1 = (7, 2)

test\_t2 = (7, 8)

L3 = []

L1 = list(test\_t1)

L2 = list(test\_t2)

for i in L1:

for j in L2:

a = (i, j)

L3.append(a)

b = (j, i)

L3.append(b)

print(L3)

**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)]**

A95.

input\_list = [('for', 24), ('Geeks', 8), ('Geeks', 30)]

L2 = []

import sys

# print(input\_list[0][1])

for i in range(0, len(input\_list)):

min = sys.maxsize

for j in input\_list:

if j[1] < min:

min = j[1]

ele = j

input\_list.remove(ele)

L2.append(ele)

print(L2)

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

**\***

**\* \***

**\* \* \***

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

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

A96.

n = 5

i = 1

while i <= n:

print('\* '\*i)

i+=1

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

**\***

**\*\***

**\*\*\***

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

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

A97.

n=5

i=1

while(i<=n):

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

i+=1

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

**\***

**\* \***

**\* \* \***

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

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

A98.

n = 5

i = 1

while i<=n:

print(' '\*(n-i) +'\* '\*i)

i+=1

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

**1**

**1 2**

**1 2 3**

**1 2 3 4**

**1 2 3 4 5**

A99.

n = 5

m = 1

for x in range(0, n):

i = 1

for y in range(0, m):

print(i, end=' ')

i+=1

m+=1

print('')

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

**A**

**B B**

**C C C**

**D D D D**

**E E E E E**

A100.

n = 5

p=65

for i in range(n):

for j in range(i+1):

print(chr(p), end=' ')

p+=1

print()