

Assignment-2

Python Mega Assignment

[PythonAssignment.md](#)

[iNeuron-Bigdata-BootCamp 2.0](#)

* Name-Surendra Kumar Yadav

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

Because they are not written in machine-readable language, Python programs need to be processed before machines can run them. Python is an interpreted language. This means that every time a program is run, its interpreter runs through the code and translates it into machine-readable byte code.

Q2. Why is Python called a dynamically typed language?

Python is both a strongly typed and a dynamically typed language. Strong typing means that variables do have a type and that the type matters when performing operations on a variable. Dynamic typing means that the type of the variable is determined only during runtime. Dynamically-typed languages are those (like Python, JavaScript) where the interpreter assigns variables a type at runtime based on the variable's value at the time.

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

Pros: Its Simple, easy to use. Its Free Object Oriented Contains lot of libraries supports ML, DL, etc Widely applicable Cons: Runtime error Poor memory management. Slow Not complete support for mobile app development

Q4. In what all domains can we use Python?

Data Science, Machine Learning, Deep Learning, AI, Data Engineering Web Application Game Development

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

Variables are a storage object which holds the data which can be used in the program a = 10

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

Using input() function user can read the input in runtime.

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

Default datatype of input() is string.

Q8. What is type casting?

Type casting is converting a variable from one datatype to another datatype.

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

We can get multiple input with delimiter and use split() function to use to store in different variables. x,y = input().split()

Q10. What are keywords?

Keywords are system defined objects which cant be used by the user as variable name and function name.

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

No we can't. Keywords are system defined objects which cant be used by the user as variable name and function name.

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

Indentation refers to the spaces at the beginning of a code line. Where in other programming languages the indentation in code is for readability only, the indentation in Python is very important. Python uses indentation to indicate a block of code.

Q13. How can we throw some output in Python?

Using print() statement.

Q14. What are operators in Python?

Operators are special characters which do one specific operation on the given variables.

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

// - Command the code which exclude from python interpreter. / - division operator returns decimal output.

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

: iNeuroniNeuroniNeuroniNeuron

```
In [6]: print(4 * 'iNeuron')
iNeuroniNeuroniNeuroniNeuron
```

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

```
In [9]: number = int(input())
if number % 2 == 0:
    print("Even")
else:
    print("Odd")
```

Odd

Q18. What are boolean operator?

The logical operators and, or and not are also referred to as boolean operators. While and as well as or operator needs two operands, which may evaluate to true or false, not operator needs one operand evaluating to true or false. Boolean and operator returns true if both operands return true

Q19. What will the output of the following?

1 or 0

0 and 0

True and False and True

1 or 0 or 0

1 or 0 -> 1 0 and 0 -> 0 True and False and True -> False 1 or 0 or 0 -> 1

Q20. What are conditional statements in Python?

Conditional statements in python are If and nested statements which used to check the multiple condition on a sequence manner and return the output.

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

if --> to check the first statement is true or false elif --> to check after first statement whether the condition(s) is true or false else --> to return/ print a statement or value when all the if and elif is failed.

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".

```
In [12]: age = int(input())
if age >= 18:
    print("I can vote")
elif age < 18:
    print("I can't vote")
else:
    print("Invalid input")
```

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]

```
In [13]: numbers = [12, 75, 150, 180, 145, 525, 50]
sum = 0
```

```

for i in numbers:
    if i % 2 == 0:
        print(i)
        sum += i

print(sum)

12
150
180
50
392

```

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

```

In [1]: x, y, z = input("Enter 3 Numbers Separated by Comma: ").split(",")
if int(x) > int(y) and int(x) > int(z):
    print(f"{x} is Greatest")
elif int(y) > int(z):
    print(f"{y} is Greatest")
else:
    print(f"{z} is Greatest")

8 is Greatest

```

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]
```

```

In [3]: numbers = [12, 75, 150, 180, 145, 525, 50]

for i in numbers:
    if i != 150 and i <=500 and i % 5 == 0:
        print(i)

75
180
145
50

```

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

String is a data type in python to store a array a of alpha numeric elements in a variable. a = 'string data type' b = "python"

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

We can access the string index by using square bracket and specifying the the position like below. python = "python: print(python[2]) -> returns t

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

```
string = "Big Data iNeuron" desired_output = "iNeuron" value = "Big Data iNeuron" print(value.split()[-1])
```

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

```

In [10]: string = "Big Data iNeuron"
desired_output = "norueNi"

value = "Big Data iNeuron"
print(value.split()[-1][::-1])

```

Q30. Reverse the string given in the above question.

```

In [12]: print(value[::-1])
print(value.split()[::-1])

norueNi ataD gib
['iNeuron', 'Data', 'Big']

```

Q31. How can you delete entire string at once?

Using 'del' command we can delete the entire string in python. Generally it will drop the variable which holds the string value.

Q32. What is escape sequence?

Escape sequence is special key which doesn't print anything however it helps to align the output based on the user requirement. Ex: \n - new line \b - backspace \t - tab

Q33. How can you print the below string?

'iNeuron's Big Data Course'

```
In [6]: print("\'iNeuron's Big Data Course\'")  
'iNeuron's Big Data Course'
```

Q34. What is a list in Python?

List is one of the key data structure in python which hold any type of data in it. List is a collection which is ordered and changeable. Allows duplicate members.

Q35. How can you create a list in Python?

List can be created in python using []. lst = [] --> empty list lst = [1,2,3] --> numeric list lst = ['a', 'bc', 'd'] --> string list lst = ['abc', 'xyz', 1] --> alphanumeric list

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

we can access the list using index.

```
In [10]: lst = [1,2,3,4,5]  
print(lst[1])  
print(lst[1:2])  
  
2  
[2]
```

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

```
In [11]: 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.

```
In [17]: print("Enter the list of numbers separated by ','")  
lst = input().split(',')  
print(len(lst))  
  
Enter the list of numbers separated by ','  
1
```

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

```
In [19]: 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?

Tuple is a collection of objects separated by commas. Same like list but tuples are immutable. Tuple is denoted by () Tuple consumes less memory compare to list. Insertion and deletion is not allowed in Tuple, allowed in List

Q41. How can you create a tuple in Python?

```
In [23]: tuple = (1,2,3,5)  
print(tuple)  
  
(1, 2, 3, 5)
```

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

```
In [26]: name = ()  
name[0] = 'SKY'  
print(name)
```

```
-----  
TypeError                                     Traceback (most recent call last)  
Input In [26], in <cell line: 2>()  
      1 name = ()  
----> 2 name[0] = 'SKY'  
      3 print(name)  
  
TypeError: 'tuple' object does not support item assignment
```

It throws an error since the tuple is immutable.

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

append method is not available in tuple. so we can concatenate two tuples and extend it.

```
In [28]: number1 = (1,2,3)  
number2 = (3,4,5)  
num = number1 + number2  
print(num)  
  
(1, 2, 3, 3, 4, 5)
```

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

```
In [30]: print(len(num))  
6
```

Q45. What are sets in Python?

Sets are collections of unique object in Python. Sets wont have duplicate values Sets are not ordered. Sets wont support indexing and ordering

Q46. How can you create a set?

We can create set using curly brackets {}. Keep in mind empty {} will result in dictionary hence there must be some value in the brackets.

```
In [42]: set_var = {1,2,3,5,5}  
print(set_var)  
  
{1, 2, 3, 5}
```

```
In [44]: my_list = [1,2,3,4,5,6,7,8,9,9]
```

```
In [45]: my_set = set(my_list)  
print(my_set)  
  
{1, 2, 3, 4, 5, 6, 7, 8, 9}
```

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

```
In [49]: my_set = set({})  
my_set.add("iNeuron")  
print(my_set)  
  
{'iNeuron'}
```

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

```
In [50]: my_set.update([1,2])  
print(my_set)  
  
{1, 2, 'iNeuron'}
```

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

We can add more than one element in a single go using update(), but using add() it's not possible.add() - helps to add a single item to the set. it will add anywhere in the set. update() - helps to add multiple items to the set. it will add the sequential order. list, tuple, sets can be added to the set using update()

Q50. What is clear() in sets?

To remove all the elements from the set, clear() function is used.clear() - to clear the elements from the set variable. it makes sets empty but not delete

Q51. What is frozen set? Q52. How is frozen set different from set?

Frozenset is similar to set in Python, except that frozensets are immutable, which implies that once generated, elements from the frozenset cannot be added or removed. This function accepts any iterable object as input and transforms it into an immutable object.

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

union() in Sets combine the two or more different sets() and eliminate the duplicate.

```
In [57]: set1 = {1,2,3,4,5}
set2 = {3,4,5,6,7}
new_set = set1.union(set2)
print(new_set)

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

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

intersection() - to return the common elements between the two sets.

```
In [60]: new_set = set1.intersection(set2)
print(new_set)

{3, 4, 5}
```

Q55. What is dictionary ibn Python?

Dictionary in Python is a collection of key values, used to store data values like a map, which, unlike other data types which hold only a single value as an element. or Dictionary used to store key values pair object in the variable. Dictionary is ordered, changeable Dictionary does not allow duplicates

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

Dictionary is having key and value pair where as all other data structures have only values in them.

Q57. How can we declare a dictionary in Python?

We can create dictionary using curly brackets {}. Understanding:- Unlike the other DataTypes, which are capable of holding only a single value in the form of an element, a Dictionary is capable of holding the key:value pair. In a Dictionary, a colon separates all the key-value pairs from each other, while a comma separates all the keys from each other.

Q58. What will the output of the following?

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

In [67]: var = {}
print(type(var))

<class 'dict'>
```

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

```
# Declaring a Dictionary
marks={"math":90,"physics":80,"chemistry":88}

# Adding a new item to the Dictionary
marks["english"] = 83

# Printing the Dictionary
print(marks)

{'math': 90, 'physics': 80, 'chemistry': 88, 'english': 83}
```

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

```
myDict = {'Name': 'S P',
          'Age': '24',
          'Gender': 'Male',
          'City': 'BLR',
          'Country': 'IN',
          'Occupation': 'Aerospace Engineer',
          'Company': 'HTS'
        }
for i,j in myDict.items():
    print(f"Key is {i} and value is {j}")
```

```
Key is Name and value is S P
Key is Age and value is 24
Key is Gender and value is Male
Key is City and value is BLR
Key is Country and value is IN
Key is Occupation and value is Aerospace Engineer
Key is Company and value is HTS
```

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

```
In [117]: Dict = {"Name": {"f_name": "SP", "l_name": "Yadav"}, "Experience": 1.5, "Organisation": "HTS"}
for i, j in Dict["Name"].items():
    print(f"Key is {i} and value is {j}")

Key is f_name and value is SP
Key is l_name and value is Yadav
```

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

get() method in python is help to get the value of the key which passed inside get()

```
In [118]: Dict = {"Name": "SP", "Experience": 1.5, "Organisation": "HTS"}
print(Dict.get("Name"))
print(Dict.get("Organisation"))

SP
HTS
```

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

items() to read both key and value from the dictionary

```
In [119]: print(Dict.items())

dict_items([('Name', 'SP'), ('Experience', 1.5), ('Organisation', 'HTS')])
```

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

```
In [120]: print(myDict)

{'Name': 'S P', 'Age': '24', 'Gender': 'Male', 'City': 'BLR', 'Country': 'IN', 'Occupation': 'Aerospace Engineer', 'Company': 'HTS'}
```

pop() used to remove the specified item in pop() from the dictionary.

```
In [121]: myDict.pop("Age")
'24'

Out[121]:
```



```
In [122]: print(myDict)

{'Name': 'S P', 'Gender': 'Male', 'City': 'BLR', 'Country': 'IN', 'Occupation': 'Aerospace Engineer', 'Company': 'HTS'}
```

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

popitem() method removes the last inserted key-value pair from the dictionary and returns it as a tuple. popitem() remove the last item from the dictionary.

```
In [124]: print(myData.popitem())

('Languages', {'good': 'python', 'better': 'sql'})
```

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

keys() method returns a view object that displays a list of all the keys in the dictionary.

```
In [127]: Dict = {"Name": "SP", "Experience": 1.5, "Organisation": "HTS"}
print(Dict.keys())

dict_keys(['Name', 'Experience', 'Organisation'])
```

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

values() is an inbuilt method in Python programming language that returns a view object. The view object contains the values of the dictionary, as a list.

```
In [129]: Dict = {"Name": "SP", "Experience": 1.5, "Organisation": "HTS"}
print(Dict.values())

dict_values(['SP', 1.5, 'HTS'])
```

Q68. What are loops in Python?

Loops are used to iterate over a block of statement multiple times.

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

There are two types of loops in Python: > for loop > while loop

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

When we know the exact number of iterations, we can use for loop. When we want the loop to run till a certain condition is true we can use while loop. for loop iterates over the range of elements and performs the operations. While loop performs some action if the condition flag is true

Q71. What is the use of continue statement?

Continue Statement skips the execution of the program block from after the continue statement and forces the control to start the next iteration. The continue keyword is used to end the current iteration in a for loop (or a while loop), and continues to the next iteration

Q72. What is the use of break statement?

break statement in Python is used to bring the control out of the loop when some external condition is triggered. break statement is put inside the loop body

Q73. What is the use of pass statement?

The pass statement is a null statement. But the difference between pass and comment is that comment is ignored by the interpreter whereas pass is not ignored.

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

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. range() function is to generate the sequence of numbers between the start and end number given, excluding the end number. By default it takes 0 if we give only one number in it.

Q75. How can you loop over a dictionary?

1. using keys 2. using .items() 3. using .keys() 4. using .values()

```
In [2]: statesAndCapitals = {
    'Gujarat': 'Gandhinagar',
    'Maharashtra': 'Mumbai',
    'Rajasthan': 'Jaipur',
    'Bihar': 'Patna'
}

for state in statesAndCapitals:
    print(state)
```

```
Gujarat
Maharashtra
Rajasthan
Bihar
```

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

```
In [1]: def factorial(n):
    if n < 0:
        return 0
    elif n == 0 or n == 1:
        return 1
    else:
        fact = 1
        while(n>1):
            fact *= n
            n -= 1
        return fact

print("Enter the Number:")
n = int(input())
print(f"Factorial of {n} is {factorial(n)}")
```

```
Enter the Number:
Factorial of 5 is 120
```

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

```
In [22]: def SI(P,R,T):
    si = (P * R * T) / 100
    return si
```

```

print("Enter the Principal:")
P = int(input())
print("Enter the rate of Interest:")
R = float(input())
print("Enter the terms(in Year):")
T = int(input())

print(SI(P,R,T))

```

Enter the Principal:
Enter the rate of Interest:
Enter the terms(in Year):
3.84

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

```

In [26]: def A(P,R,t):
    return P * (R/100)**t

print(A(P,R,T))

```

2.0971520000000004e-06

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

```

In [4]: def isPrime(n):
    if n == 2:
        return True
    if n % 2 == 0:
        return False

    for i in range(2,int(n**0.5)+1,2):
        if n % i == 0:
            return False
        else:
            return True

    return True

number = int(input())
print(isPrime(number))

```

True

Q80. Write a Python program to check Armstrong Number.

```

In [6]: def isArmstrong(n):

    sum = 0

    if n == 0:
        return True
    else:
        while n > 0:
            rem = n // 2
            sum += pow(rem, 3)
            n = n % 2

    return sum

number = int(input())
if number == isArmstrong:
    print(f'The Number {number} is Armstrong Number')
else:
    print(f'The Number {number} is not Armstrong Number')

```

The Number 99 is not Armstrong Number

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

```

In [9]: def Fibonacci(n):
    if n<= 0:
        print("Incorrect input")
    elif n == 1:
        return 0
    elif n == 2:
        return 1
    else:
        return Fibonacci(n-1)+Fibonacci(n-2)

```

```
print(Fibonacci(7))
```

```
8
```

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

```
In [12]: def swap_list(newList):
    size = len(newList)
    temp = newList[0]
    newList[0] = newList[size - 1]
    newList[size - 1] = temp

    return newList

newList = [15, 12, 35, 17, 9, 56, 29]
print(swap_list(newList))

[29, 12, 35, 17, 9, 56, 15]
```

```
In [13]: def swap_list(newList):
    newList[0], newList[-1] = newList[-1], newList[0]

    return newList

newList = [15, 12, 35, 17, 9, 56, 29]
print(swap_list(newList))

[29, 12, 35, 17, 9, 56, 15]
```

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

```
In [15]: my_list = [1, 2, 3, 4, 5]

print(my_list)

temp = my_list[len(my_list)-1]
my_list[len(my_list)-1] = my_list[0]
my_list[0] = temp

print(my_list)

[1, 2, 3, 4, 5]
[5, 2, 3, 4, 1]
```

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

```
In [17]: my_list = [1, 2, 3, 4, 5]

my_list.sort(reverse=True)
print(my_list)

print("Enter the nth Big number to find")
nth = int(input())
print(my_list[nth-1])

[5, 4, 3, 2, 1]
Enter the nth Big number to find
```

```
1
```

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

```
In [18]: my_list = [1, 2, 3, 4, 5]
sum = 0

for i in my_list:
    sum = sum + i

print(sum)

15
```

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

```
In [21]: def isPalindrome(s):
    if s == s[::-1]:
        return f"{s} is palindrome"
    return f"{s} is not palindrome"
```

```
s = "dad"
isPalindrome(s)

Out[21]: 'dad is palindrome'
```

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

```
In [22]: myStr = 'Big data Ccourse'
new_str = myStr.replace('c', '')
print(new_str)

Big data Course
```

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

```
In [28]: my_sentence = "Big Data Bootcamp 2.0 Assignments"

my_list = my_sentence.split(' ')
print(my_list)

find = input('Enter the substring')
print(find)

if find in my_list:
    print(f'{find} is present in the sentence')
else:
    print(f'{find} is not present in the sentence')

['Big', 'Data', 'Bootcamp', '2.0', 'Assignments']

2.0
2.0 is present in the sentence
```

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

```
In [34]: def string_greater_than_k(k, str):

    string = []
    text = str.split(" ")
    for x in text:
        if len(x) > k:
            string.append(x)
    return string

k = 3
str = "Big Data Bootcamp 2.0 Assignments"
print(string_greater_than_k(k, str))

['Data', 'Bootcamp', 'Assignments']
```

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

```
In [36]: dict1 = {'A' : [1, 3, 5, 4],
             'B' : [4, 6, 8, 10],
             'C' : [6, 12, 4, 8],
             'D' : [5, 7, 2]}
print("The original dictionary is : ",dict1)

# Using List comprehension, values() and sorted()
res = list(sorted({ele for val in dict1.values() for ele in val}))

# print result
print("The unique values list is : ", res)

The original dictionary is :  {'A': [1, 3, 5, 4], 'B': [4, 6, 8, 10], 'C': [6, 12, 4, 8], 'D': [5, 7, 2]}
The unique values list is :  [1, 2, 3, 4, 5, 6, 7, 8, 10, 12]
```

Q91. Write a Python program to merge two dictionary.

```
In [37]: def Merge(dict1, dict2):
    return(dict2.update(dict1))

# Driver code
dict1 = {'a': 10, 'b': 8}
```

```

dict2 = {'d': 6, 'c': 4}

# This returns None
print(Merge(dict1, dict2))

# changes made in dict2
print(dict2)

None
{'d': 6, 'c': 4, 'a': 10, 'b': 8}

```

Q92. Write a Python program to convert a list of tuples into dictionary.

```

In [38]: Input : [('Sachin', 10), ('MSD', 7), ('Kohli', 18), ('Rohit', 45)]
          Output : {'Sachin': 10, 'MSD': 7, 'Kohli': 18, 'Rohit': 45}

def Convert(tup, di):
    di = dict(tup)
    return di

tups = [('Sachin', 10), ('MSD', 7), ('Kohli', 18), ('Rohit', 45)]
dictionary = {}
print (Convert(tups, dictionary))

{'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.

```

In [39]: Input: list = [9, 5, 6]
          Output: [(9, 729), (5, 125), (6, 216)]

lst = [9, 5, 6]

result_list = [(val, pow(val, 3)) for val in lst]
print(result_list)

[(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)]

```

```

In [42]: test_tuple1 = (7, 2)
          test_tuple2 = (7, 8)

result_list = [(tuple1, tuple2) for tuple1 in test_tuple1 for tuple2 in test_tuple2]
result_list = result_list + [(tuple1, tuple2) for tuple1 in test_tuple2 for tuple2 in test_tuple1]
print(result_list)

[(7, 7), (7, 8), (2, 7), (2, 8), (7, 7), (7, 2), (8, 7), (8, 2)]

```

Q95. Write a Python program to sort a list of tuples by second item.

```

Input : [('452', 10), ('256', 5), ('100', 20), ('135', 15)]
Output : [('256', 5), ('452', 10), ('135', 15), ('100', 20)]

```

```

In [52]: tuple = [('452', 10), ('256', 5), ('100', 20), ('135', 15)]
          print("Original Tuple List :" ,tuple)

def Sort(tuple):
    # Sorts in Ascending order
    tuple.sort(key = lambda a: a[1])
    return tuple

print("Sorted Tuple List:" ,Sort(tuple))

Original Tuple List : [('452', 10), ('256', 5), ('100', 20), ('135', 15)]
Sorted Tuple List: [('256', 5), ('452', 10), ('135', 15), ('100', 20)]

```

Q96. Write a python program to print below pattern.

```

*
* *
* * *
* * * *
* * * * *

```

```
In [2]: n = int(input('Enter the Value'))
for i in range(n+1):
    print('*' * i)
```

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

Q97. Write a python program to print below pattern.

```
*
```

```
**
```

```
***
```

```
****
```

```
*****
```

```
In [1]: n = int(input('Enter the Value'))
for i in range(n+1):
    print('*'*(n-i) + '*'*(i+1))
```

```
*
```

```
**
```

```
***
```

```
****
```

```
*****
```

```
*****
```

Q98. Write a python program to print below pattern.

```
*
```

```
* *
```

```
* * *
```

```
* * * *
```

```
* * * * *
```

```
In [19]: n = int(input('Enter the Row: '))
for i in range(n+1):
    print('*'*(n-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
```

```
In [17]: n = int(input('Enter the Row: '))
for i in range(1,n):
    for j in range(1,i+1):
        print(j, 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
```

```
In [15]: val = 65  
  
n = int(input('Enter the value of Row:'))  
for i in range(n):  
    for j in range(i+1):  
        print(chr(val), end=' ')  
    val+=1  
    print('\n')
```

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

The End
