Module Code & Module Title

CS4051NI Fundamentals of Computing

Assessment Weightage & Type

60% Individual Coursework

Year and Semester

2021 Spring

Student Name : Punam Thapa Magar

Group : C3

London Met ID:20048968

College ID : NPO1CP4S210273

Assignment Due Date : 21th September- 2021

Assignment Submission Date : 21th September-2021

I confirm that I understand my coursework needs to be submitted online via Google classroom under the relevant module page before the deadline in order for my assignment to be accepted and marked. I am fully aware that late submission will be treated as non-submission and a marks of zero will be awarded.

Table of Contents

List of Figures

Question no 1

Question no 1.py - D:/Computing exam/Question no 1.py (3.9.6)

File Edit Format Run Options Window Help

```
L = [9, 2, 1, 7, 5]
print(L)

i = 1
while i < len(L):
    key = L[i]
    j = i - 1
    while j >= 0 and key < L[j]:
        L[j + 1] = L[j]
        j = j - 1
    else:
        L[j + 1] = key
        i = i + 1
else:
    print(L)
```

Figure 1: Screen shots from Question no 1

Output

IDLE Shell 3.9.6

File Edit Shell Debug Options Window Help

```
Python 3.9.6 (tags/v3.9.6:db3ff76, Jun 28 2021, 15:26:21) [MSC v.1929 64 bit (AM
D64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
========== RESTART: D:/Computing exam/Question no 1.py ==========
[9, 2, 1, 7, 5]
[1, 2, 5, 7, 9]
>>>
```

Figure 2: Screen shots of output

Question no 2

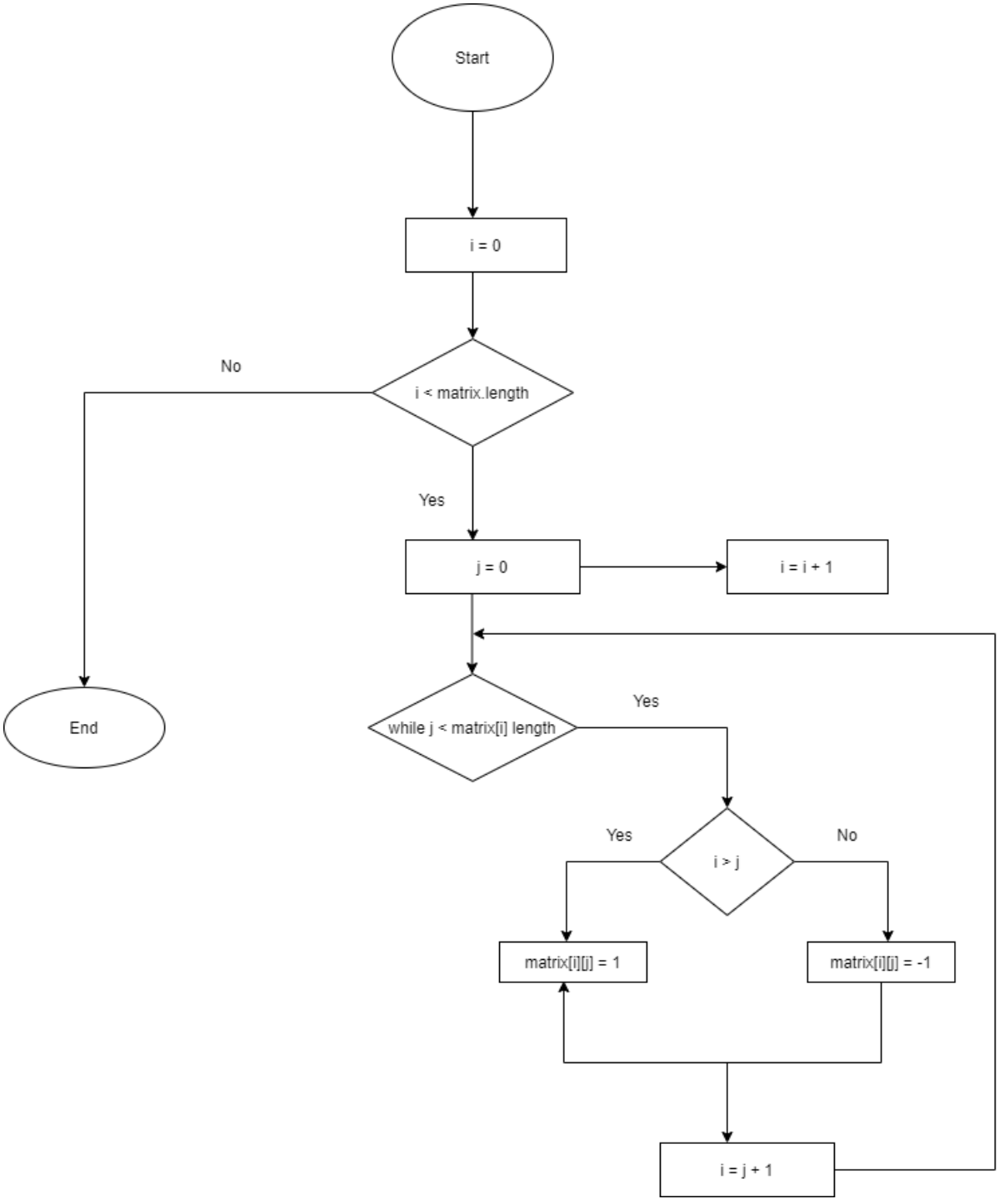


Figure 3: Flow chart

A variable i is characterized and the worth of i is assigned out as 0. Then, at that point, the while circle checks if the worth of i is not exactly the quantity of things in the 2D list names as matrix. While the worth i is not exactly the quantity of things in the 2D list the circle runs on the up and up, another variable j is characterized and its qualities is alloted as 0. Presently another while circle was made which runs , while the value of j is not exactly the length of lines of the framework. Then, at that point, the program checks the condition where if the worth of is I is more noteworthy than that of j, it replaces the worth of the record I and j at the network with 1. At the point when I isn't more prominent than j then the worth at list of I and j in the given network is supplanted with - 1. Similarly, the value of j is expanded by 1, after the execution of conditions, the worth of I is expanded by 1 when the internal circle reaches a conclusion.

Question no 3 (a)

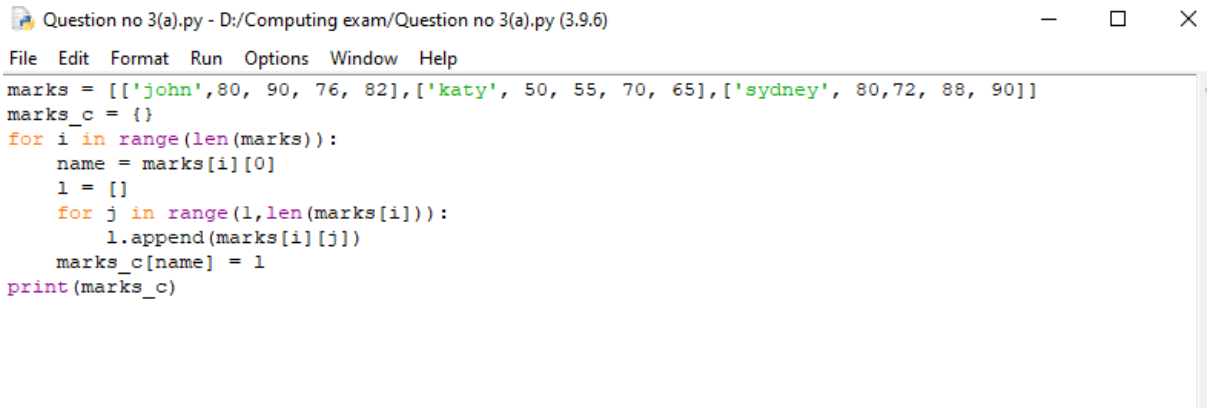


Figure 4: Screen shots of corrected code

Output

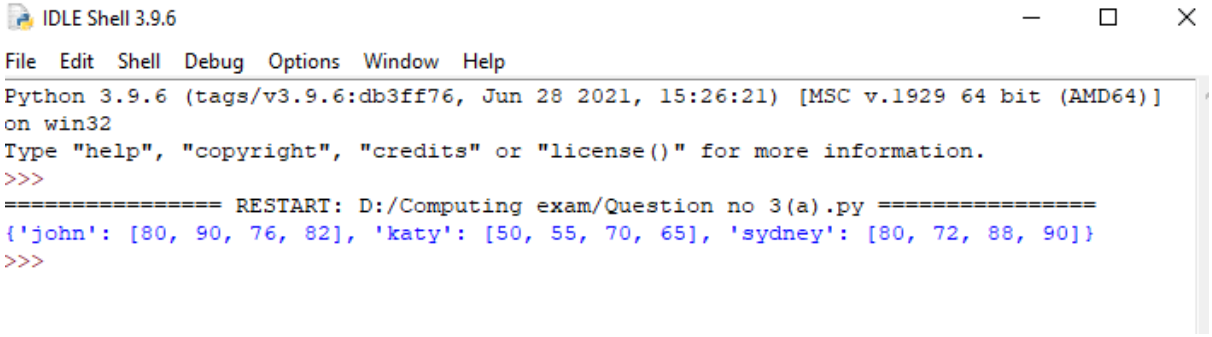


Figure 5: Screen shots of corrected code output

The list of errors to be fixed in the program are:

i. for j in range(2, len(marks[i])):
ii. l.append(marks[i][j])
iii. marks_c[name] = l

Question no 3 (b)

Question no 3x(b).py - D:\Computing exam\Question no 3x(b).py (3.9.6)

File Edit Format Run Options Window Help

```
class Person:
    def __init__(self, name, address, number):
        """Constructor of Person"""
        self.name = name
        self.address = address
        self.number = number

    def get_Person(self):
        """Returns the name of a person"""
        return self.name

    def get_all_detail(self):
        """Returns all the details of person"""
        return "HI! My name is " + self.name + " and I live at " + self.address

    def get_contact(self):
        """Returns the contact of the person"""
        return self.number


class Employee(Person):
    def __init__(self, name, address, number, salary, department):
        """Constructor of Employee"""
        Person.__init__(self, name, address, number)
        self.salary = salary
        self.department = department

    def get_contact(self):
        """Checks if the number is none and returns the contact of the
        person"""
        if self.number == None:
            return "Does not have a number"
        else:
            return self.number

    def change_number(self, new_number):
        """method for CHANGING the number of the person"""
        self.number = new_number

    def change_address(self, addr):

        """method for CHANGING the address of the person """
        self.address = addr

foo = Person("Ram", "Kamalpokhari", "9812121212")
print(foo.get_all_detail())
print(foo.get_contact())
bar = Employee("Rabi", "Gaushala", "9836699636", "40000", "HR")
print(bar.get_all_detail())
bar.change_address("Koteshwor")
print(bar.get_all_detail())
```

Figure 6: Screen shots of corrected codes

Output

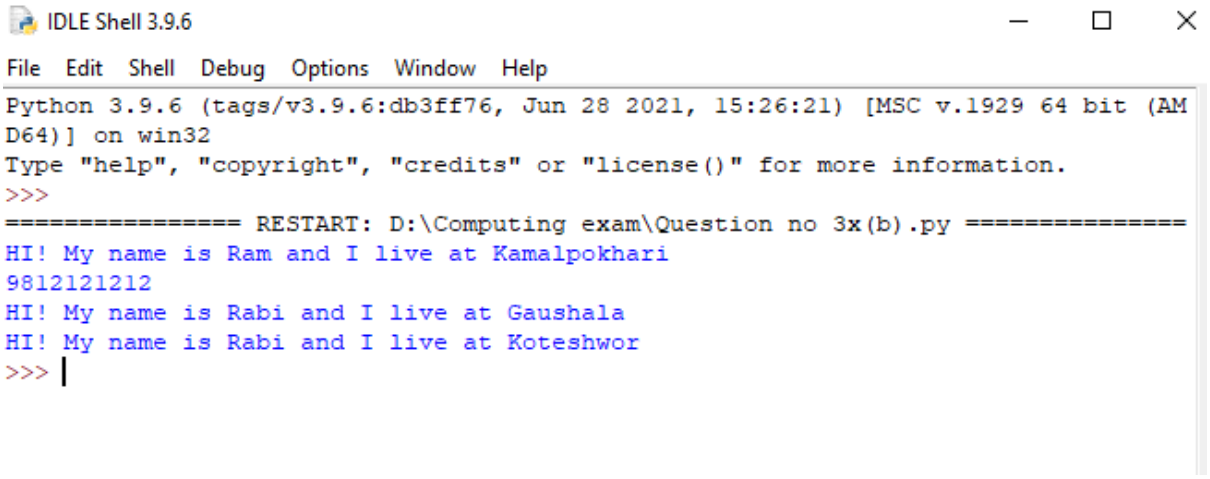


Figure 7: Screen shots of corrected code output

The list of errors to be fixed in the program are given below:

1. def__init__(self, name, address, number)
2. self.name == name
3. return name
4. class Employee:
5. Person__init__(self, name, address)
6. self.number = None:
7. self.addr = addr
8. print(foo.get_all_details()),print(bar.get_all_details())
   print(bar.get_all details()

Question no 4 (Answer)

Python has been utilized worldwide for various fields making websites, artificial intelligence and considerably more. In any case, to make the entirety of this potential, data assumes a vital part which implies that this data ought to be put away effectively and the access to it must be convenient. So to accomplish this we use something called Data Structure.

Collection in python are fundamentally compartment data types, namely lists, sets, tuples, dictionary. They have various attributes dependent on the revelation and the utilization.

The collection data types in python are described below:

List

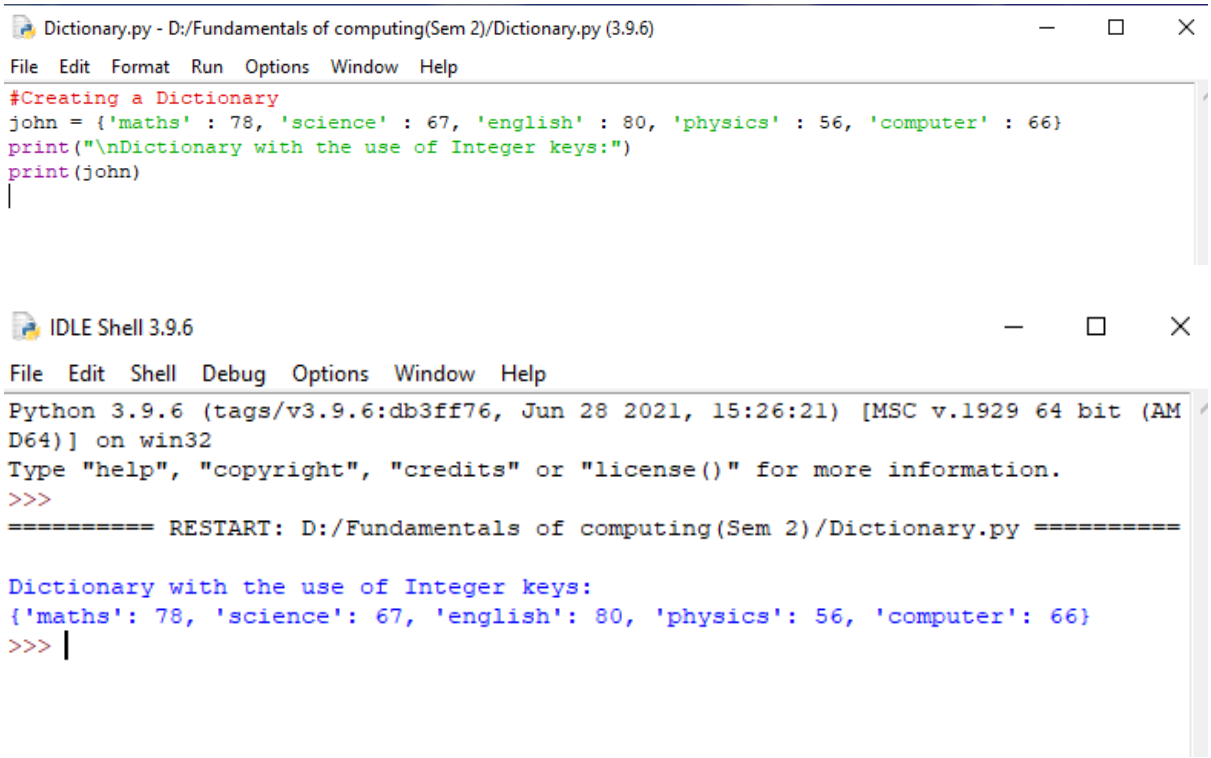
A list is an ordered sequence of information, accessible by index which is denoted by square brackets [ ]. Likewise, a list contains usually of the same element types and can contain another list to form a multi-dimensional list. List can be cut, concatenated indexed and update individual things in the list. List are mutable so their structure and content can be changed. Thus, list can be utilized properly while working with stacks, queues, sorting items, matrixes and working with enormous measure of data.

List can be created as follows:

```
bookname = []
authorname = []
quantity = []
price = []
```

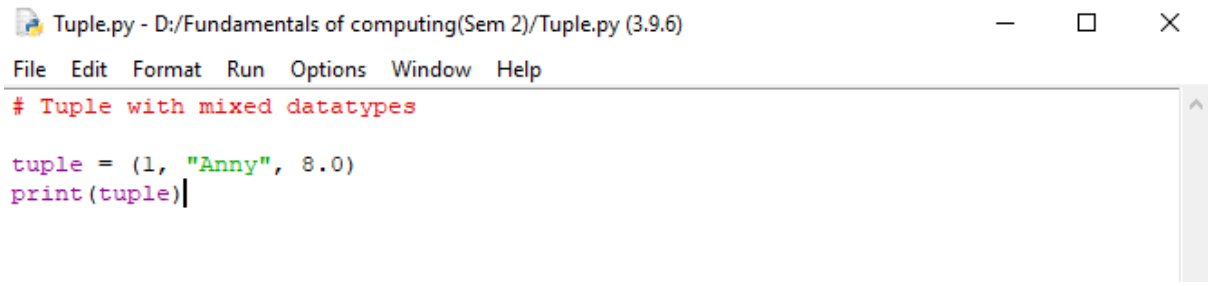
Dictionary

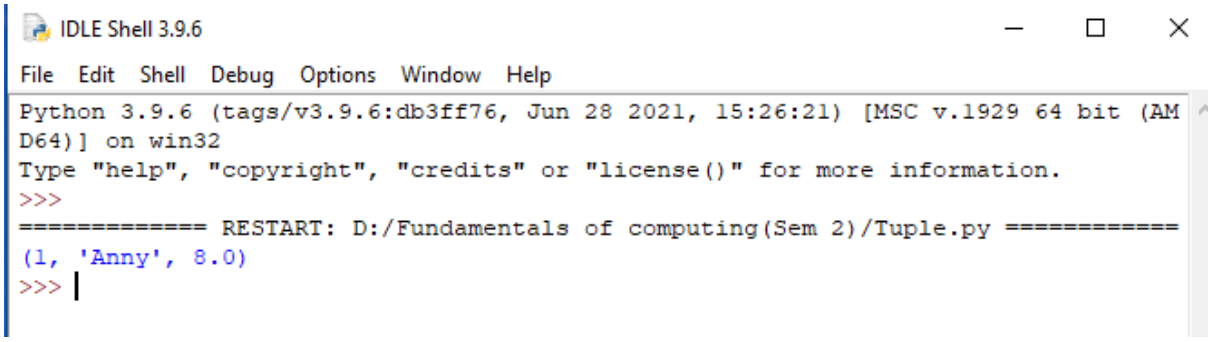
In Python, a Dictionary can be made by setting an arrangement of components inside wavy { } supports, isolated by 'comma'. Word reference holds a couple of qualities, one being the Key and the other relating pair component being its key:value. Qualities in a word reference can be of any information type and can be copied, while keys can't be rehashed and should be changeless.



Tuples

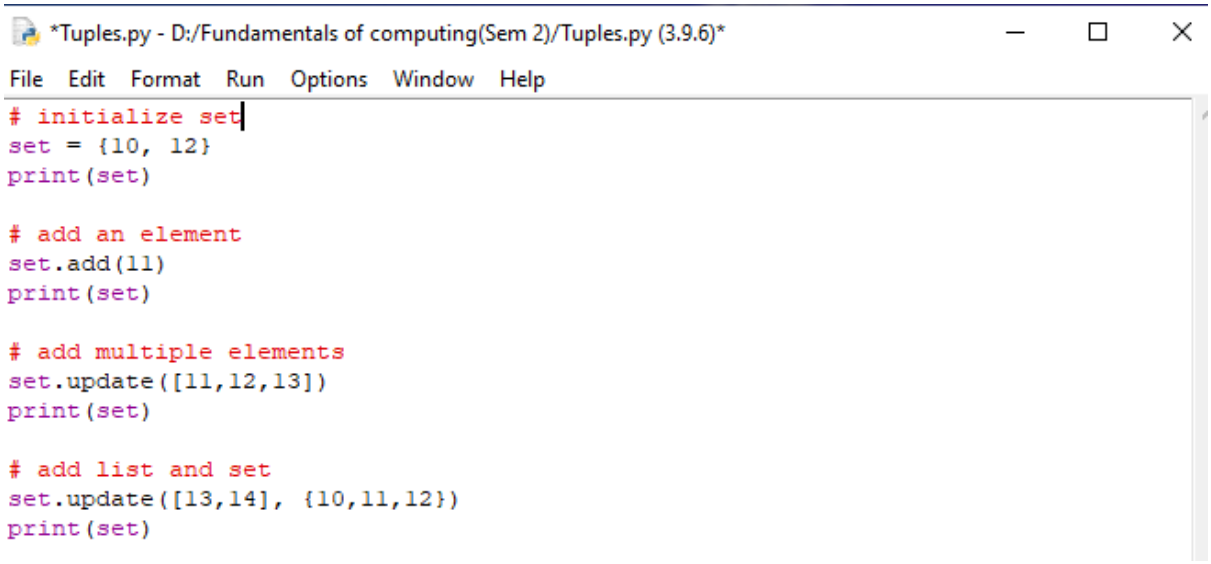
Tuples are utilized to store various items in a solitary variable. It is one of the four inherent data types in python used to store collection of information, the other three are List, Set, and Dictionary, all with various characteristics and uses. Moreover, a tuple is an assortment which is ordered, allow duplicates values and unchangeable, which is basically composed with round brackets.

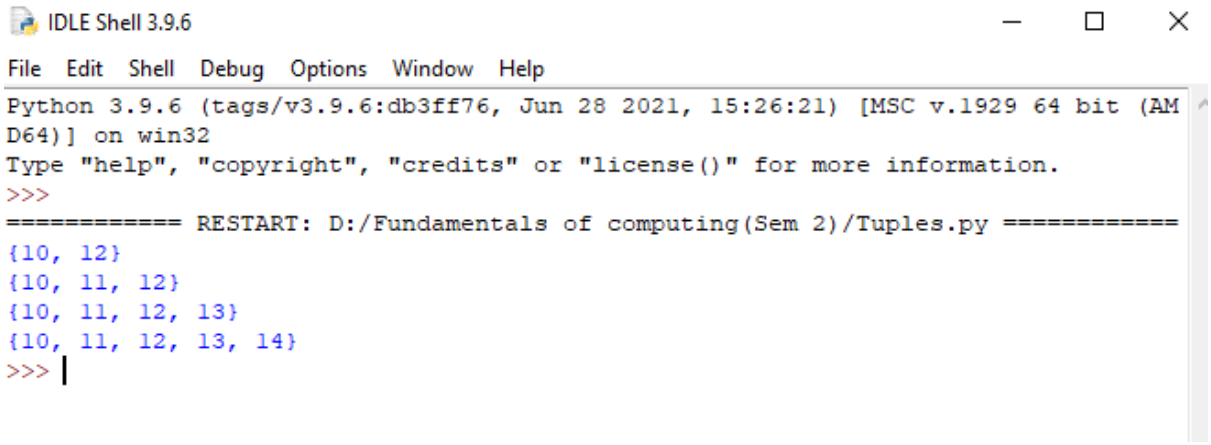




Sets

Mathematically a set is a collection of items not in any particular order. A python set is similar to this mathematical definition. However, the elements in the python set cannot be duplicates and are immutable but the set as a whole is mutable. Likewise, there is no index attached to any elements in a python set. So they do not support any indexing or slicing operation. A set is created by using the set ( ) function or placing all the elements with a pair of curly braces





Besides, in this as we talked about over that dictionary are compound datatypes with keys and values likewise they are variable. So in the given table we have different films with their relating ID, rent cost and quantity. Subsequently, it can be made by giving the components (for instance: film ID, film name, lease value, amount) inside in the wavy sections by isolating it with comma. Then, at that point, after word reference holds the pair of qualities that is known as keys:esteem. Also esteems can be the components in the table, for example, M001 Pulp Fiction $5 30, M002 Lord of The Rings $2.5 10, M003 The Revenant $4 20.

Question no 5(a)

Question no 5(a).py - D:/Computing exam/Question no 5(a).py (3.9.6)

File Edit Format Run Options Window Help

```
#Write a program that takes input from the user a string of numbers (eg."24453")
#then all the numbers from the string must be put in a list.
#The program then computes the sum, maximum and minimum from the list of numbers

number = input("Enter a string of numbers:")


n = list(number)
sum_ = 0
max_ = n[0]
min_ = n[0]

for i in range(len(n)):
    m = n[i]
    sum_ = sum_+int(m)

    if(max_ < m):
        max_ = m

    if(min_ > m):
        min_ = m


print("The sum of numbers is:", sum_)
print("The maximum from the list is:", max_)
print("The minimum from the list is:", min_)
```

Figure 8: Screen shots of code

Output:

IDLE Shell 3.9.6

File Edit Shell Debug Options Window Help

```
Python 3.9.6 (tags/v3.9.6:db3ff76, Jun 28 2021, 15:26:21) [MSC v.1929 64 bit (AM
D64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
================ RESTART: D:/Computing exam/Question no 5(a).py ================
Enter a string of numbers:24453
The sum of numbers 18
The maximum from the list 5
The minimum from the list 2
>>>
```

Figure 9: Screen shots of output

Question no 5 (b)

Question no 5(b).py - D:\Computing exam\Question no 5(b).py (3.9.6)

File Edit Format Run Options Window Help

```
 #Write a program that takes 2 words as input from the user and prints out a list
#containing the letters which the 2 words have in common. For example,
#if word1 is "english"  and word2 is "nepali", the output should be ["l", "e", "i", "n"].

word1 = input("Enter First word:")
word2 = input("Enter another word:")
x1 = set(word1)
x2 = set(word2)
Y = list(x1 & x2)

print("Common letters:" , Y)
```

Figure 10: Screen shots of code

Output:

IDLE Shell 3.9.6

File Edit Shell Debug Options Window Help

```
Python 3.9.6 (tags/v3.9.6:db3ff76, Jun 28 2021, 15:26:21) [MSC v.1929 64 bit (AM
D64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
================ RESTART: D:/Computing exam/Question no 5(b).py ================
Enter First word:english
Enter another word:nepali
Common letters: ['i', 'e', 'n', 'l']
>>>
```

Figure 11: Screen shots of output