

Islington college
(इरिलिङ्टन कलेज)

Module Code & Module Title

CC4051N FUNDAMENTALS OF COMPUTING

Assessment Weightage & Type

60% Individual Coursework

Year and Semester

2020-2021 Summer

Student Name: SALAJ SUBEDI

London Met ID: 20049007

College ID: np01cp4s210311

Assignment Due Date: 21st September 2021

Assignment Submission Date: 21st 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 submissions will be treated as non-submission and a mark of zero will be awarded.

Table of contents

Question 1.....	3
Question 2.....	4
Question 3a.....	6
Question 3b.....	7
Question 4.....	10
Question 5a.....	11
Question 5b.....	12

Table of figures

Figure 1: question1 code.....	3
Figure 2: question 1 output.....	3
Figure 3: Flow chart for operations of question no.2	5
Figure 4: question 3a code.....	6
Figure 5: question 3a output.....	6
Figure 6: question 3b code.....	8
Figure 7: question 3b output.....	9
Figure 8: question 5a code.....	11
Figure 9: question 5a output.....	11
Figure 10: question 5b code.....	12
Figure 11: question 5b output.....	12

Question 1.**Solution**

```

1.py - D:/COLLEGE/ALL_MODULES/Second Semester/module-fundamentals of computing/EXAM 21st SE...
File Edit Format Run Options Window Help
L = [9,2,1,7,5]
i = 1
#loop while i is less than the no.of elements in list L
while i < len(L):
    key = L[i]
    j = i - 1 #set j to be i - 1

    while j >= 0 and key < L[j]: #initialize and loop while j is less than or equal to 0
        L[j+1] = L[j]
        j = j - 1 #reducing value of J to terminate program when needed

    L[j+1] = key #placing all the updated elements on the list.
    i = i + 1 #adding value of i by 1 for continuation of loop

print(L)

```

Figure 1: question1 code

```

1.py - D:/COLLEGE/ALL_MODULES/Second Semester/module-fundamentals of computing/EXAM 21st SE...
File Edit Format Run Options Window Help
L = [9,2,1,7,5]
i = 1
#loop while i is less than the no.of elements in list L
while i < len(L):
    key = L[i]
    j = i - 1 #set j to be i - 1

    while j >= 0 and key < L[j]: #initialize and loop while j is less than or equal to 0
        L[j+1] = L[j]
        j = j - 1 #reducing value of J to terminate program when needed

    L[j+1] = key #placing all the updated elements on the list.
    i = i + 1 #adding value of i by 1 for continuation of loop

print(L)

```

```

IDLE Shell 3.9.5
File Edit Shell Debug Options Window Help
Python 3.9.5 (tags/v3.9.5:0a7dcdbd, May 3 2021, 17:27:52) [MSC v.1928 64 bit (AMD64)]
Type "help", "copyright", "credits" or "license()" for more information.
>>>
= RESTART: D:/COLLEGE/ALL_MODULES/Second Semester/module-fundamentals of computing/EXAM 21st SEPTEMBER/py files/1.py
[1, 2, 5, 7, 9]
>>>

```

Figure 2: question 1 output

Question 2.**Solution**

In the given code the user is asked to input a 3x3 scaled matrix. The program runs and at first prints all the rows that were initially provided in the matrix. The user then sets the start value of *i* to 0 and provides a condition where if *i* is smaller than the length of matrix the value of *j* will be set to 0, also further conditioned when the value of *j* is smaller than the length of matrix with value of *i*, In this program the `matrix[i][j]` is 1, *i,j* indicated the position of element where the value of 1 is to be kept and updated if the above conditions are not satisfied otherwise if value of *i* is smaller than the length of matrix with value *j* the `matrix[i][j]` is updated with -1 the loop continues until the initial condition i.e. `i < length of matrix` is true and lastly prints ----- and once the line is printed and condition is false it executes the last command and prints the new and updated 3x3 matrix with values of 1 and -1 that have been updated from the above conditions

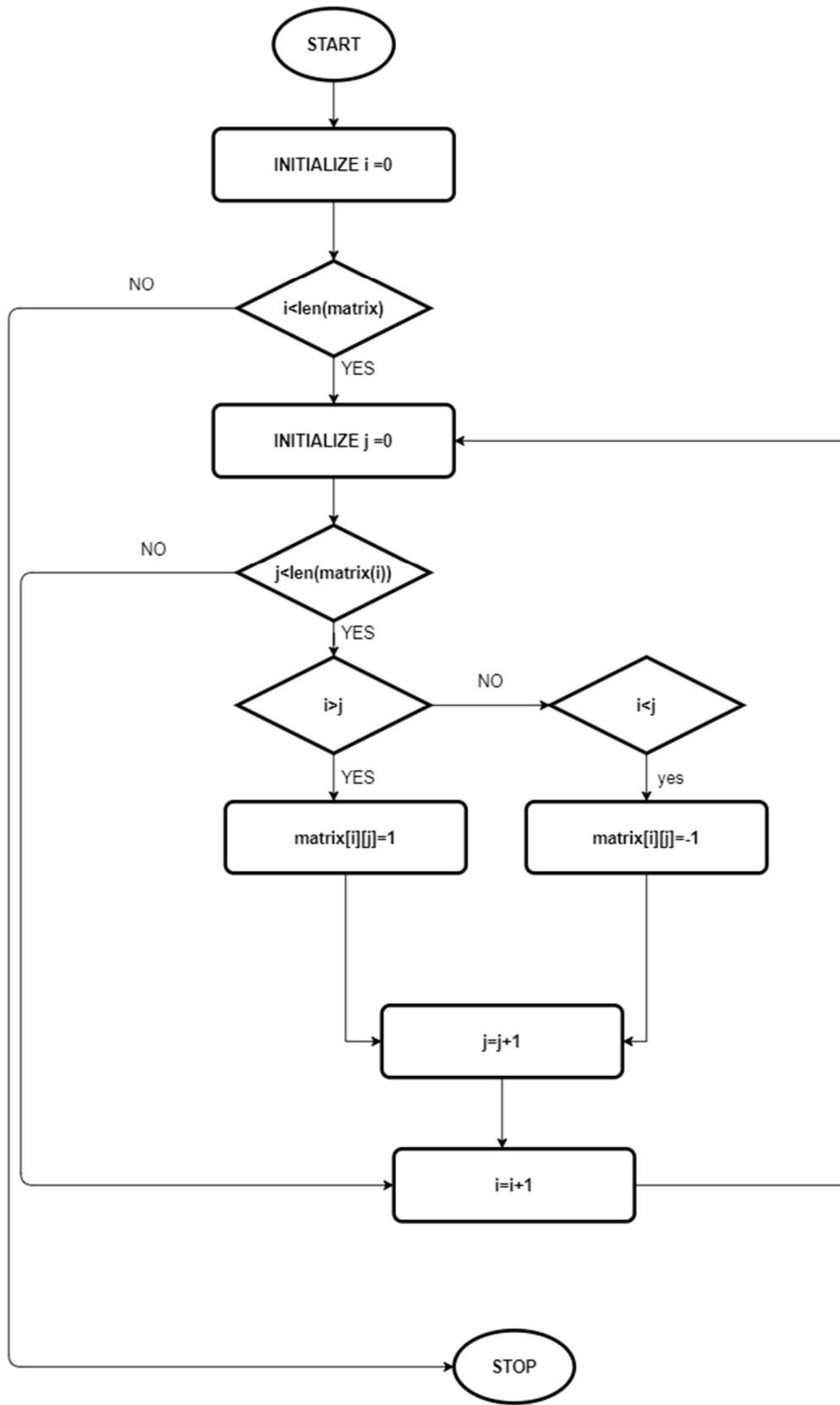
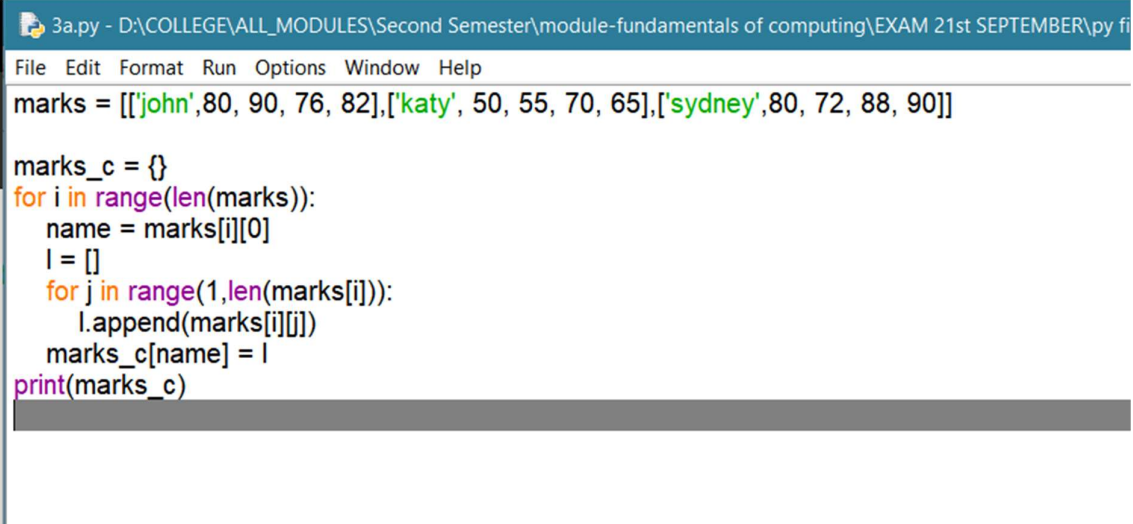


Figure 3: Flow chart for operations of question no.2

Question 3a.**Solution**

List of errors fixed.

1. Removed repeated and additional lines of code i.e. "print(marks_c)"
2. Altered the starting of "j" loop to start from 1 instead of 2
3. Updated
4. the empty list "l" with values from the list marks
5. Updated empty dictionary "marks_c" to contain names as the keys and rest of the elements in each list within the nested list as its values



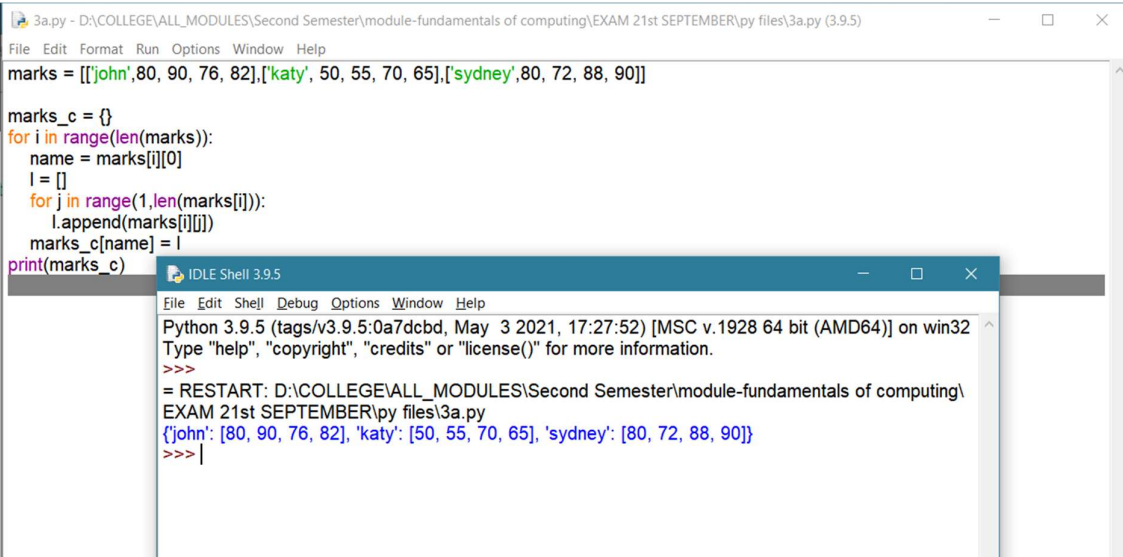
```

3a.py - D:\COLLEGE\ALL_MODULES\Second Semester\module-fundamentals of computing\EXAM 21st SEPTEMBER\py fi
File Edit Format Run Options Window Help
marks = [['john',80, 90, 76, 82],['katy', 50, 55, 70, 65],['sydney',80, 72, 88, 90]]

marks_c = {}
for i in range(len(marks)):
    name = marks[i][0]
    l = []
    for j in range(1,len(marks[i])):
        l.append(marks[i][j])
    marks_c[name] = l
print(marks_c)

```

Figure 4: question 3a code



```

3a.py - D:\COLLEGE\ALL_MODULES\Second Semester\module-fundamentals of computing\EXAM 21st SEPTEMBER\py files\3a.py (3.9.5)
File Edit Format Run Options Window Help
marks = [['john',80, 90, 76, 82],['katy', 50, 55, 70, 65],['sydney',80, 72, 88, 90]]

marks_c = {}
for i in range(len(marks)):
    name = marks[i][0]
    l = []
    for j in range(1,len(marks[i])):
        l.append(marks[i][j])
    marks_c[name] = l
print(marks_c)

```

Python 3.9.5 (tags/v3.9.5:0a7dcdbd, May 3 2021, 17:27:52) [MSC v.1928 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
= RESTART: D:\COLLEGE\ALL_MODULES\Second Semester\module-fundamentals of computing\EXAM 21st SEPTEMBER\py files\3a.py
{'john': [80, 90, 76, 82], 'katy': [50, 55, 70, 65], 'sydney': [80, 72, 88, 90]}
>>> |

Figure 5:question 3a output

Question 3b.**Solution****List of errors fixed**

1. Added self-keyword where necessary to represent instance of a particular class.
2. Removed '==' and replaced with '=' where value was being assigned to variables
3. Removed '=' and replaced with '==' where comparison operators were being used to compare.
4. Corrected the spelling of functions and variable used
5. Corrected proper syntax of subclass Employee to represent class person.
Added missing brackets where necessary.

```
3.b.py - D:/COLLEGE/ALL_MODULES/Second Semester/module-fundamentals of computing/EXAM 21st SEPTEMBER/py files/3.b.py (3.9.5)
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

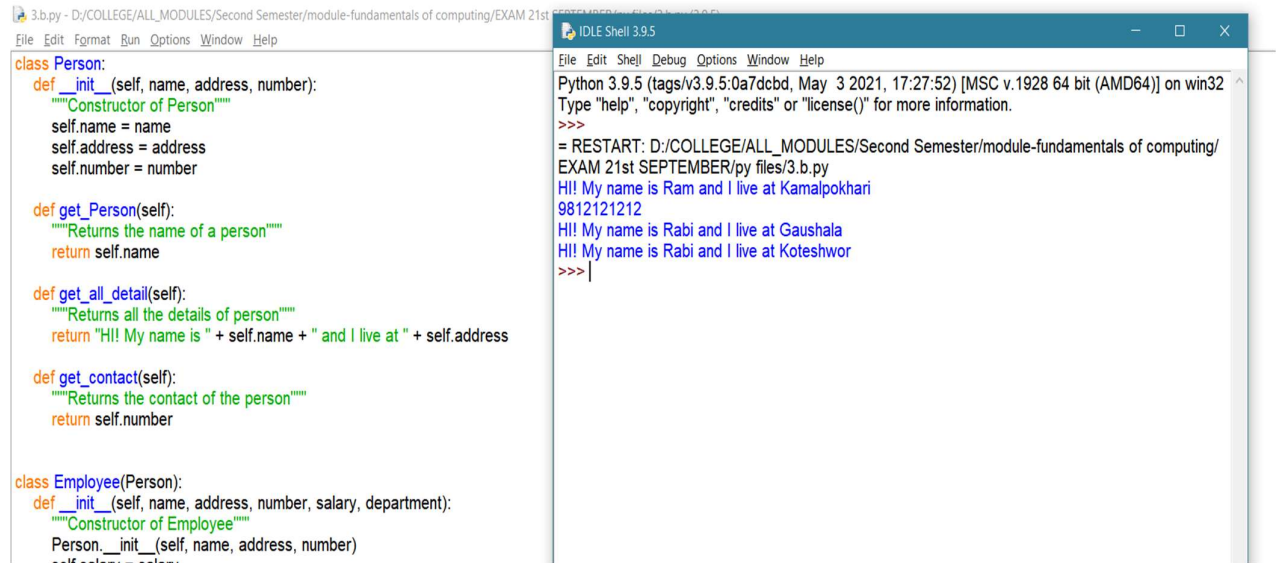
    def get_contact(self):
        """Checks if the number is none and returns the contact of the person"""
        if self.number == None:
            return "Does not have a number"
        else:
            return self.number

    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: question 3b code



```
3.b.py - D:/COLLEGE/ALL_MODULES/Second Semester/module-fundamentals of computing/EXAM 21st SEPTEMBER/py files/3.b.py
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

IDLE Shell 3.9.5
File Edit Shell Debug Options Window Help
Python 3.9.5 (tags/v3.9.5:0a7dcbd, May 3 2021, 17:27:52) [MSC v.1928 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
= RESTART: D:/COLLEGE/ALL_MODULES/Second Semester/module-fundamentals of computing/
EXAM 21st SEPTEMBER/py files/3.b.py
Hi! My name is Ram and I live at Kamalpokhari
9812121212
Hi! My name is Rabi and I live at Gaushala
Hi! My name is Rabi and I live at Koteswhor
>>>
```

Figure 7: question 3b output

Question 4.**Solution**

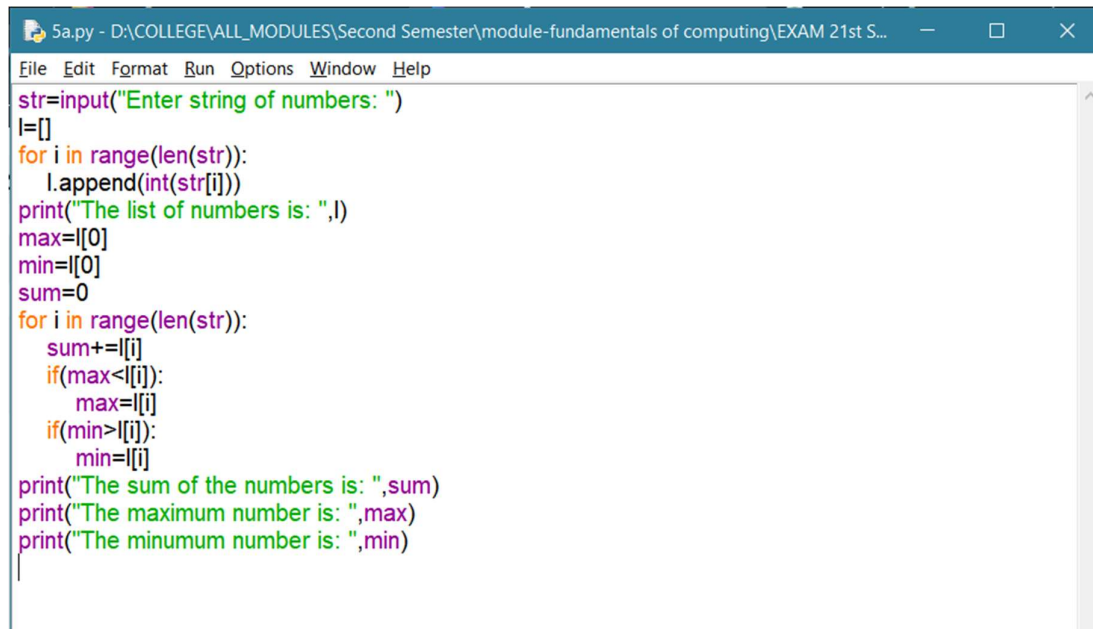
There are many data collection types in python to handle different groups of data in specified and efficient format.

They are

1. Lists: In Python programming, a list is created by placing all the items (elements) inside square brackets [], separated by commas. It can have any number of items and they may be of different types (integer, float, string etc.). A list can also have another list as an item. This is called a nested list. e.g.: `list1 = ["abs", 34, True, 40, "male"]`
2. Tuples: Tuples are used to store multiple items in a single variable in ordered sequence. Tuple is one of 4 built-in data types in Python used to store collections of data, the other 3 are List, Set, and Dictionary, all with different qualities and usage. A tuple is a collection which is ordered and unchangeable. Tuples are written with round brackets. `tuple1 = ("abs", 34, True, 40, "male")`
3. Sets: Sets are used to store multiple items in a single variable. A set is a collection which is unordered, unchangeable and unindexed and ignores repeated values. Sets are written with curly brackets.
`thisset = {"apple", "banana", "cherry", "apple"}`
`print(thisset)===== {'banana', 'cherry', 'apple'}`
4. Dictionaries: Dictionaries are used to store data values in key:value pairs. dictionary is a collection which is ordered*, changeable and does not allow duplicates. Dictionaries are written with curly brackets, and have keys and values: `thisdict = {"brand": "Ford", "model": "Mustang", "year": 1964}`

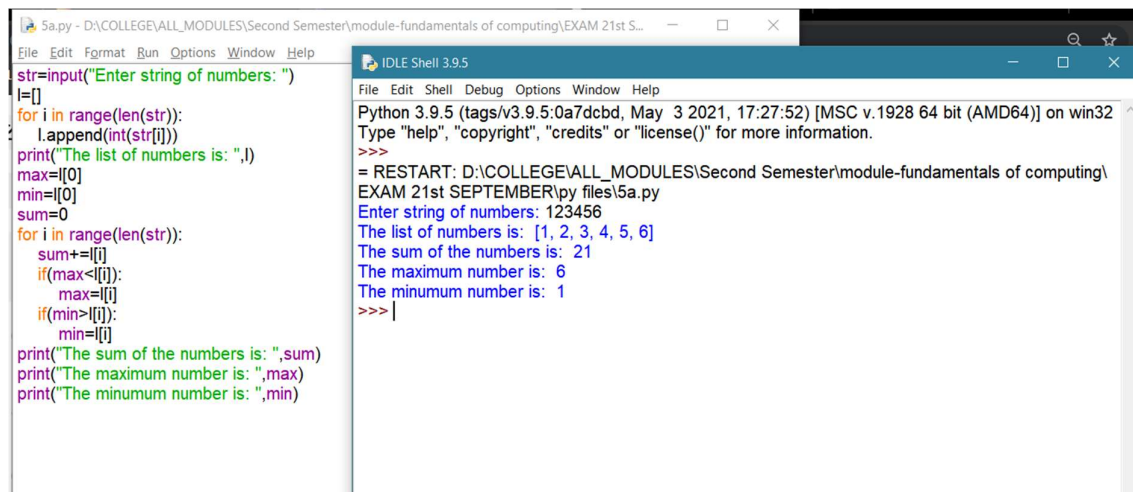
Above the collection datatypes storing the movie details in a dictionary would be optimal decision as dictionary stored data as keys and values so each keys can represent a movie id, movie name, rent price, quantity as the heading and the respective values can be stored accordingly. E.g

`movies={"Movie ID": "M001", "MOVIE NAME": "Pulp Fiction", "Rent price": "$5", "Quantity": "30"}` this way a movie can be stored and displayed which gives the users correct understanding and reading of the movie and its details. Further dictionaries are ordered, changeable and don't allow duplicated too so in that case the admin can adjust the quantity rate and the id or names of books as per the changes in real life also if the admins enter duplicate or double values the program auto deletes the repeated values and keeps from creating duplication there by saving space and economy. Above conclusions provided claim that dictionaries as a collection data type to store movies would be convenient, efficient and economically wise in the part of the admin needing it.

Question 5a.**Solution**

```
5a.py - D:\COLLEGE\ALL_MODULES\Second Semester\module-fundamentals of computing\EXAM 21st S...
File Edit Format Run Options Window Help

str=input("Enter string of numbers: ")
l=[]
for i in range(len(str)):
    l.append(int(str[i]))
print("The list of numbers is: ",l)
max=l[0]
min=l[0]
sum=0
for i in range(len(str)):
    sum+=l[i]
    if(max<l[i]):
        max=l[i]
    if(min>l[i]):
        min=l[i]
print("The sum of the numbers is: ",sum)
print("The maximum number is: ",max)
print("The mininum number is: ",min)
```

Figure 8: question 5a code

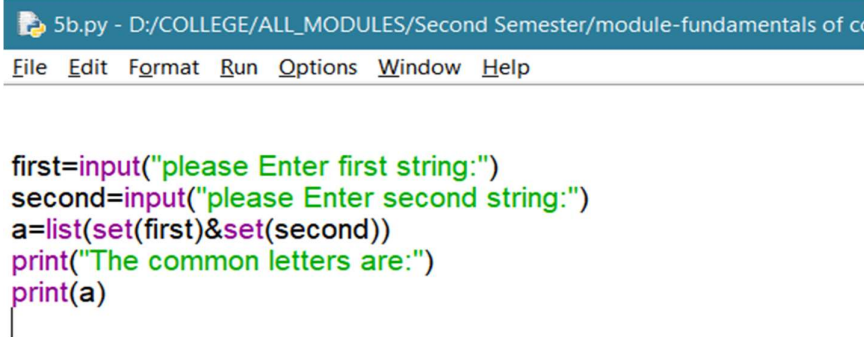
```
5a.py - D:\COLLEGE\ALL_MODULES\Second Semester\module-fundamentals of computing\EXAM 21st S...
File Edit Format Run Options Window Help

str=input("Enter string of numbers: ")
l=[]
for i in range(len(str)):
    l.append(int(str[i]))
print("The list of numbers is: ",l)
max=l[0]
min=l[0]
sum=0
for i in range(len(str)):
    sum+=l[i]
    if(max<l[i]):
        max=l[i]
    if(min>l[i]):
        min=l[i]
print("The sum of the numbers is: ",sum)
print("The maximum number is: ",max)
print("The mininum number is: ",min)
```

```
IDLE Shell 3.9.5
File Edit Shell Debug Options Window Help

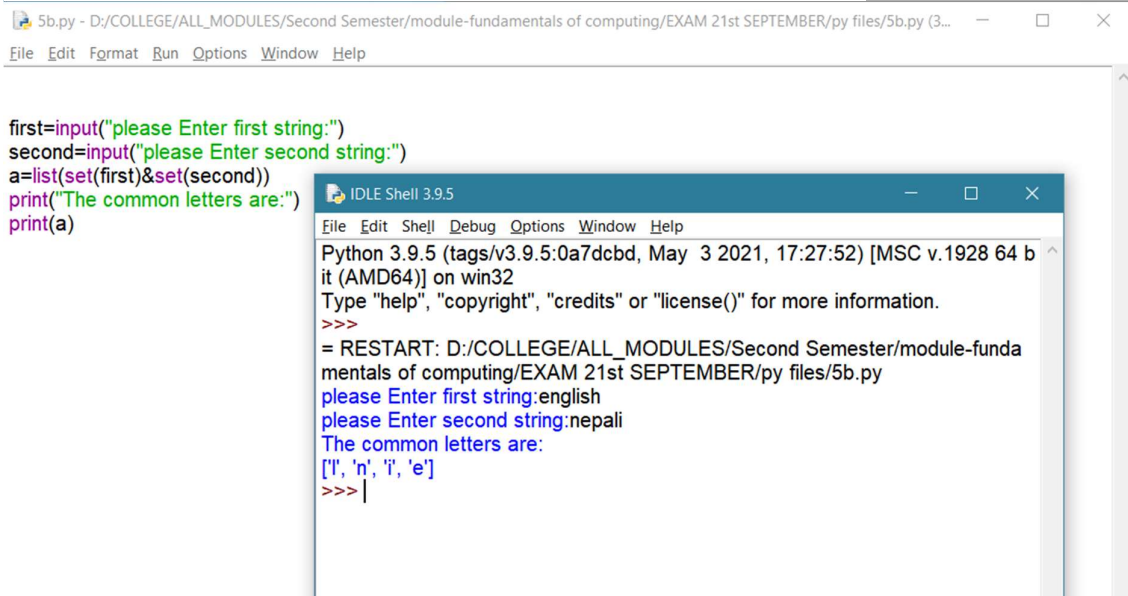
Python 3.9.5 (tags/v3.9.5:0a7dcbbd, May 3 2021, 17:27:52) [MSC v.1928 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
= RESTART: D:\COLLEGE\ALL_MODULES\Second Semester\module-fundamentals of computing\
EXAM 21st SEPTEMBER\py files\5a.py
Enter string of numbers: 123456
The list of numbers is: [1, 2, 3, 4, 5, 6]
The sum of the numbers is: 21
The maximum number is: 6
The minimum number is: 1
>>> |
```

Figure 9: question 5a output

Question 5b.**Solution**

```
5b.py - D:/COLLEGE/ALL_MODULES/Second Semester/module-fundamentals of co
File Edit Format Run Options Window Help

first=input("please Enter first string:")
second=input("please Enter second string:")
a=list(set(first)&set(second))
print("The common letters are:")
print(a)
```

Figure 10: question 5b code

```
Sb.py - D:/COLLEGE/ALL_MODULES/Second Semester/module-fundamentals of computing/EXAM 21st SEPTEMBER/py files/5b.py (3...
File Edit Format Run Options Window Help

first=input("please Enter first string:")
second=input("please Enter second string:")
a=list(set(first)&set(second))
print("The common letters are:")
print(a)
```

Python 3.9.5 (tags/v3.9.5:0a7dcdb, May 3 2021, 17:27:52) [MSC v.1928 64 b
it (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
= RESTART: D:/COLLEGE/ALL_MODULES/Second Semester/module-funda
mentals of computing/EXAM 21st SEPTEMBER/py files/5b.py
please Enter first string:english
please Enter second string:nepali
The common letters are:
['l', 'h', 'i', 'e']
>>> |

Figure 11: question 5b output