Name: Md. Rashid Aziz

Roll No: 2003013

Batch : C11

Experiment 4

AIM:

To study list, set, tuple, dictionary

Theory:

•Sets:

Mathematical Definition: a set is a well-defined collection of distinct objects, typically called elements or members.

Python's built-in data type 'Set' has following characteristics:

* Sets are unordered
* Set elements are unique. Duplicate elements are not allowed
* A set itself is mutable i.e. you can add or remove elements from set
* Elements contained in the sets must be of immutable type. (strings, tuples, Numeric)
* Sets can have different type of elements that is they can be heterogeneous in nature

A set can be created using two ways:

1. Built-in set ( ) function: It uses iterable

   Syntax: x = set(<iter>)

   Here, argument <iter> is an iterable. It could be of type 'list', 'tuple' or 'string'.

2. With curly braces { }: Make group of immutable elements

   Syntax: x = { <obj1>, <obj2>, <obj3> ... <objn> }

Given sets x1 and x2, Possible operations on sets are

1. Union:

> Union operation combines elements of x1 and x2

> The method can be called as x1.union(x2)

> The operator (|) is also used for union operation as x1 | x2

2. Intersection:

> Intersection operation identifies common elements between x1 and x2

> The method can be called as x1.intersection(x2)

> The operator (&) is also used for union operation as x1 & x2

3. Difference:

> Difference operation identifies elements present in x1 but not in x2.

> The method can be called as x1.difference(x2)

> The operator (-) is also used for union operation as x1 - x2

4. Symmetric Difference:

> Symmetric Difference operation identifies elements present in x1 or in x2 but not in both.

> The method can be called as x1.symmetric_difference(x2)

> The operator (^) is also used for union operation as x1 ^ x2

• Dictionary:

A dictionary represents a group of elements arranged in the form of key-value pairs.

In the dictionary, the first element is considered as 'key' and immediate next element is taken as its 'value'.

The key and values are separated by colon (:)

All the key-value pairs in a dictionary are inserted in curly braces

Example: dict = { 'Name':'Sam', 'ID':200, 'Salary':20000 }

The methods to retrieve or to manipulate content of dictionaries are:

> clear(): Removes all the elements from the dictionary

> copy(): Returns a copy of the dictionary

> fromkeys(): Returns a dictionary with the specified keys and value

> get(): Returns the value of the specified key

> items(): Returns a list containing a tuple for each key value pair

> keys(): Returns a list containing the dictionary's keys

> pop(): Removes the element with the specified key

> popitem(): Removes the last inserted key-value pair

> setdefault(): Returns the value of the specified key. If the key does not
exist: insert the key, with the specified value

> update(): Updates the dictionary with the specified key-value pairs

> values(): Returns a list of all the values in the dictionary

•Tuple:

A tuple in Python is similar to a list. The difference between the two is that we cannot change the elements of a tuple once it is assigned whereas we can change the elements of a list.

A tuple is created by placing all the items (elements) inside parentheses (), separated by commas.

Example: tup = (1, 2, 3, 4, ‘python’)

Functions to process tuples:

> count(): Returns the number of times the given element appears in the tuple.

Syntax:

tuple.count(element)

> index(): Returns the first occurrence of the given element from the tuple.

Syntax:

tuple.index(element, start, end)

Parameters:

element: The element to be searched.

start (Optional): The starting index from where the searching is started

end (Optional): The ending index till where the searching is done

> len(): Returns the number of elements in the tuple

> max(): Returns the biggest element in the tuple

> min(): Returns the smallest element in the tuple

> sorted(): Sorts the elements of the tuple into asecding order

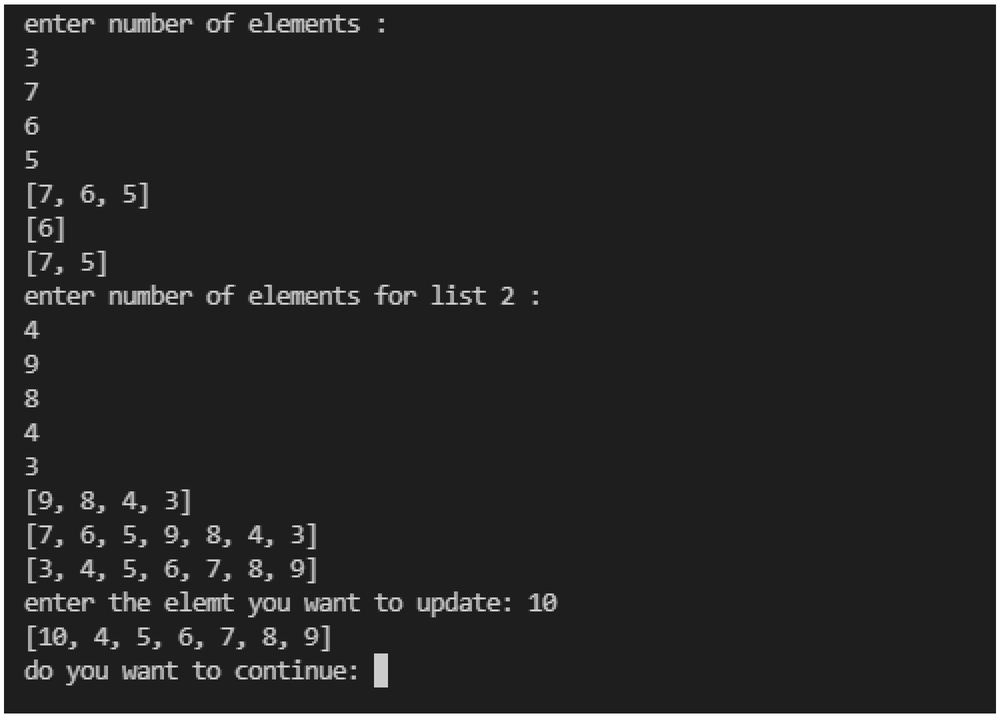
Program 1:

```
ch = 'y'
item = []
while ch == 'y':
    print("enter number of elements :");
    n = int(input());
```

```
size = n
while n > 0:
    a = int(input());
    item.append(a);
    n = n -1
print(item);
itemOdd = []
itemEven = []
for i in range(0, size):
    if item[i] % 2 == 0:
        itemEven.append(item[i])
    else:
        itemOdd.append(item[i])
print(itemEven)
print(itemOdd)
item2 = [];
print("enter number of elements for list 2 :");
n = int(input());
size2 = n
while n > 0:
    a = int(input());
    item2.append(a);
    n = n -1
print(item2)
item.extend(item2)
```

```
    print(item)
    item.sort()
    print(item)
    a = int(input("enter the elemt you want to update: "))
    item[0] = a;
    print(item)
    ch = input("do you want to continue: ");
```

Output:



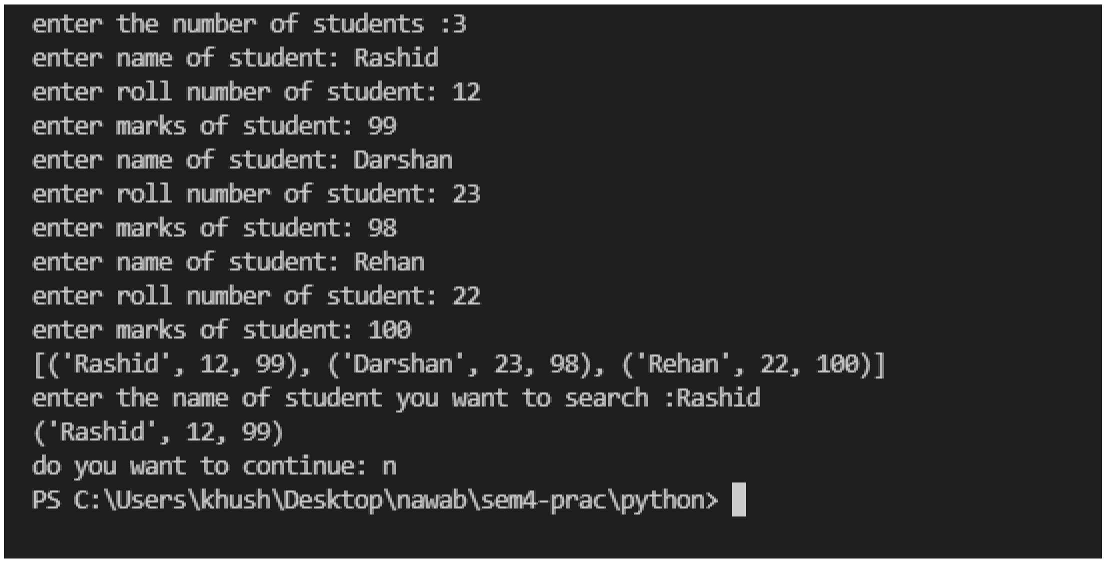
```
enter number of elements :
3
7
6
5
[7, 6, 5]
[6]
[7, 5]
enter number of elements for list 2 :
4
9
8
4
3
[9, 8, 4, 3]
[7, 6, 5, 9, 8, 4, 3]
[3, 4, 5, 6, 7, 8, 9]
enter the elemt you want to update: 10
[10, 4, 5, 6, 7, 8, 9]
do you want to continue:
```

Program 2:

```
ch = 'y'
student = []
while ch == 'y':
    n = int(input("enter the number of students :"))
```

```
for i in range(0, n):
    name = input("enter name of student: ");
    rn = int(input("enter roll number of student: "));
    marks = int(input("enter marks of student: "));
    tup = (name, rn, marks)
    student.append(tup)

print(student)
name = input("enter the name of student you want to search :");
for i in range(0, n):
    if student[i][0] == name:
        print(student[i]);
ch = input("do you want to continue: ");
```

Output:

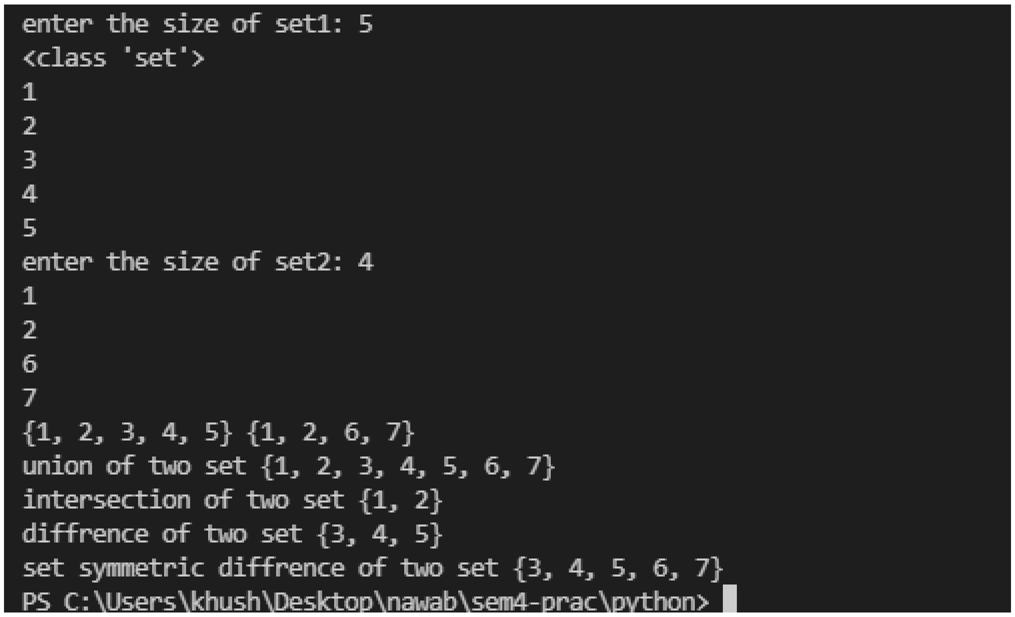


```
enter the number of students :3
enter name of student: Rashid
enter roll number of student: 12
enter marks of student: 99
enter name of student: Darshan
enter roll number of student: 23
enter marks of student: 98
enter name of student: Rehan
enter roll number of student: 22
enter marks of student: 100
[('Rashid', 12, 99), ('Darshan', 23, 98), ('Rehan', 22, 100)]
enter the name of student you want to search :Rashid
('Rashid', 12, 99)
do you want to continue: n
PS C:\Users\khush\Desktop\nawab\sem4-prac\python>
```

Program 3:

```
n1 = int(input("enter the size of set1: "))
set1 = set()
print(type(set1))
for i in range(0, n1):
    a = int(input())
    set1.add(a);
n2 = int(input("enter the size of set2: "))
set2 = set()
for i in range(0, n2):
    a = int(input())
    set2.add(a);

print(set1, set2)
print("union of two set", set1.union(set2));
print("intersection of two set", set1.intersection(set2));
print("diffrence of two set", set1.difference(set2))
print("set symmetric diffrence of two set", set1.symmetric_difference(set2))
```

Output:



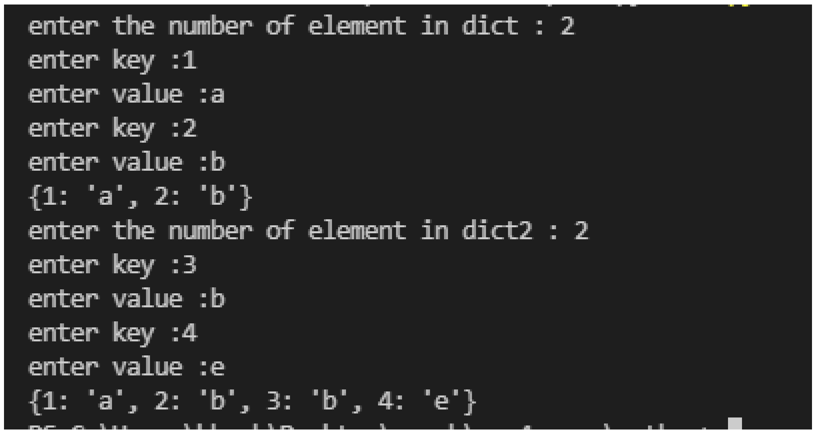
```
enter the size of set1: 5
<class 'set'>
1
2
3
4
5
enter the size of set2: 4
1
2
6
7
{1, 2, 3, 4, 5} {1, 2, 6, 7}
union of two set {1, 2, 3, 4, 5, 6, 7}
intersection of two set {1, 2}
diffrence of two set {3, 4, 5}
set symmetric diffrence of two set {3, 4, 5, 6, 7}
PS C:\Users\khush\Desktop\nawab\sem4-prac\python>
```

Program 3:

```
n = int(input("enter the number of element in dict : "))
dic = {}
for i in range(0, n):
    a = int(input("enter key :"))
    b = input("enter value :")
    dic[a] = b

print(dic)
# dict1 = OrderedDict(sorted(dic.items()))
# print(dict1)



n = int(input("enter the number of element in dict2 : "))
dic2 = {}
```

```
for i in range(0, n):
    a = int(input("enter key :"))
    b = input("enter value :")
    dic2[a] = b

dic.update(dic2)
print(dic)
```

Output:



```
enter the number of element in dict : 2
enter key :1
enter value :a
enter key :2
enter value :b
{1: 'a', 2: 'b'}
enter the number of element in dict2 : 2
enter key :3
enter value :b
enter key :4
enter value :e
{1: 'a', 2: 'b', 3: 'b', 4: 'e'}
```