

**ITA6017 - PYTHON PROGRAMMING**  
**ASSIGNMENT 3**

**REG NO: 22MCA0223**

**NAME: KAMRAN ANSARI**

## Question 1

Define a function which can print a dictionary where the keys are numbers between 1 and 20 (both included) and the values are square of keys.

Hints:

Use dict[key]=value pattern to put entry into a dictionary. Use \*\* operator to get power of a number. Use range() for loops.

## Code

```
def print_dict_table():  
    table_dict = {}  
  
    for i in range(1, 21):  
        table_dict[i] = i**2  
  
    print(table_dict)  
  
print_dict_table()
```

## Output

```
deadmercury@DESKTOP-QUKDS0U MINGW64 /e/LEARNING/vit_mca (main)  
$ C:/Python311/python.exe e:/LEARNING/vit_mca/ITA6017_PYTHON/DA/DA_3/Q1.py  
{1: 1, 2: 4, 3: 9, 4: 16, 5: 25, 6: 36, 7: 49, 8: 64, 9: 81, 10: 100, 11: 121, 12: 144, 13: 169, 14: 196, 15: 225, 16: 256, 17: 289, 18: 324, 19: 361, 20: 400}
```

## Question 2

Given a value 'n', build a dictionary with first 'n' terms in the sequence as keys and the list of their divisors as their values. For example, if 'n' is 8, then the expected output is: {1 : [1], 3: [1,3], 6: [1,2,3,6], 10:[1,2,5,10], 15:[1,3,5,15], 21:[1,3,7,21], 28:[1,2,4,7,14,28], 36:[1,2,3,4,6,8,9,12,18,36]}

Print the dictionary in ascending order of key. Use pprint function for printing dictionary in sorted order.

Syntax for pprint is pprint(dictionary name)

Include the line "from pprint import pprint" in the top of your program

Input Format First line contains the number of terms in the sequence, n

Output Format A dictionary with the terms of 'index sum sequence' as key and their divisors as values. Print the dictionary in ascending order of key.

## Code

```
from pprint import pprint

def find_divisors(number):
    divisors = []

    for i in range(1, (number // 2) + 1):
        if number % i == 0:
            divisors.append(i)

    divisors.append(number)

    return divisors
```

```
print("Enter n:")
n = int(input())

sequence = []

print("Enter sequence:")
for i in range(n):
    sequence.append(int(input()))

dictionary = {}

for number in sequence:
    dictionary[number] = find_divisors(number)

pprint(dictionary)
```

## Output

```
deadmercury@DESKTOP-QUKDS0U MINGW64 /e/LEARNING/vit_mca (main)
$ C:/Python311/python.exe e:/LEARNING/vit_mca/ITA6017_PYTHON/DA/DA_3/Q2.py
Enter n:
8
Enter sequence:
1
3
10
21
15
6
28
36
{1: [1],
 3: [1, 3],
 6: [1, 2, 3, 6],
10: [1, 2, 5, 10],
15: [1, 3, 5, 15],
21: [1, 3, 7, 21],
28: [1, 2, 4, 7, 14, 28],
36: [1, 2, 3, 4, 6, 9, 12, 18, 36]}
```

```
$ C:/Python311/python.exe e:/LEARNING/vit_mca/ITA6017_PYTHON/DA/DA_3/Q2.py
Enter n:
10
Enter sequence:
3
4
13
60
100
1
5
39
45
33
{1: [1],
 3: [1, 3],
 4: [1, 2, 4],
 5: [1, 5],
13: [1, 13],
33: [1, 3, 11, 33],
39: [1, 3, 13, 39],
45: [1, 3, 5, 9, 15, 45],
60: [1, 2, 3, 4, 5, 6, 10, 12, 15, 20, 30, 60],
100: [1, 2, 4, 5, 10, 20, 25, 50, 100]}
```

### Question 3

Frequency Tables:

Keys are the objects for which we want to count the frequency, and the Values are the frequencies. Import Harry Potter Movies dataset from Kaggle (the character.csv dataset) and count the frequency of each house present in the dataset.

Hints:

- Create an empty dictionary that will contain the frequency table.
- Loop through the list of houses, and if the key for a house is already present in the frequency table, we add 1 to its value.
- Otherwise, create a key for the current house and map it to value 1 (it's one because we encounter this element for the first time).

### Code

```
import csv

freqTable = {}

with open("characters.csv", mode="r") as file:
    csvFile = csv.reader(file)

    for characters in csvFile:
        house = characters[4]

        if house:
            if house in freqTable:
                freqTable[house] += 1
            else:
```

```
freqTable[house] = 1

for house, freq in freqTable.items():
    print(house, freq)
```

## Output

```
deadmercury@DESKTOP-QUKDS0U MINGW64 /e/LEARNING/vit_mca/ITA6017_PYTHON/DA/DA_3 (main)
$ python Q3.py
Gryffindor 31
Slytherin 20
Ravenclaw 12
Hufflepuff 8
Beauxbatons Academy of Magic 2
Durmstrang Institute 2
```

## Question 4

While John got ready for the office a shopping list was given by his son and daughter. While returning home John decides to buy the items from a shop. He decides to buy the items common in both the list, then the items listed only by his son and at last the items listed only by his daughter. Write a program to help him in achieving the task with sets operations.

## Code

```
sonList = []
daughterList = []

print("Enter number of items in son's list:")
numSon = int(input())

print("Enter products:")
```

```
for i in range(numSon):
    sonList.append(input())

print("Enter number of items in daughter's list:")
numDaughter = int(input())

print("Enter products:")
for i in range(numDaughter):
    daughterList.append(input())

sonSet = set(sonList)
daughterSet = set(daughterList)
similarItems = sonSet.intersection(daughterSet)

print("Son's Shopping list:", sonSet)
print("Daughter's Shopping list:", daughterSet)
print("Common items to buy first:", similarItems)
print("Remaining son's items:", sonSet - similarItems)
print("Remaining daughter's items:", daughterSet - similarItems)
```



## Output

```
deadmercury@DESKTOP-QUKDS0U MINGW64 /e/LEARNING/vit_mca (main)
$ C:/Python311/python.exe e:/LEARNING/vit_mca/ITA6017_PYTHON/DA/DA_3/Q4.py
Enter number of items in son's list:
5
Enter products:
Milk
Biscuit
Crackers
Polish
Car Toy
Enter number of items in daughter's list:
6
Enter products:
Purse
Biscuit
Notebook
Pen
Milk
Whitener
Son's Shopping list: {'Milk', 'Polish', 'Biscuit', 'Crackers', 'Car Toy'}
Daughter's Shopping list: {'Purse', 'Milk', 'Whitener', 'Biscuit', 'Pen', 'Notebook'}
Common items to buy first: {'Milk', 'Biscuit'}
Remaining son's items: {'Polish', 'Crackers', 'Car Toy'}
Remaining daughter's items: {'Notebook', 'Whitener', 'Purse', 'Pen'}
```

```
deadmercury@DESKTOP-QUKDS0U MINGW64 /e/LEARNING/vit_mca (main)
$ C:/Python311/python.exe e:/LEARNING/vit_mca/ITA6017_PYTHON/DA/DA_3/Q4.py
Enter number of items in son's list:
4
Enter products:
Bag
Whitener
Purse
Tissue
Enter number of items in daughter's list:
3
Enter products:
Taser
Watch
Bag
Son's Shopping list: {'Tissue', 'Purse', 'Bag', 'Whitener'}
Daughter's Shopping list: {'Taser', 'Bag', 'Watch'}
Common items to buy first: {'Bag'}
Remaining son's items: {'Tissue', 'Purse', 'Whitener'}
Remaining daughter's items: {'Taser', 'Watch'}
```

## Question 5

A marketing company has branch in a set of cities 'S'. The company sends three of their sales man to various cities in set 'S'. In the middle of the year, help the manager to find out the cities that are already visited by the sales men and the cities that are yet to be visited.

### Code

```
print("Enter number of cities available:")
no_of_cities = int(input())

cities = set()

print("Enter city names:")
for i in range(no_of_cities):
    cities.add(input())

cities_visited = set()

print("Enter number of cities visited by first salesman:")
cities_first = int(input())

print("Enter city names:")
for i in range(cities_first):
    cities_visited.add(input())

print("Enter number of cities visited by second salesman:")
cities_second = int(input())
```

```
print("Enter city names:")
for i in range(cities_second):
    cities_visited.add(input())

print("Enter number of cities visited by third salesman:")
cities_third = int(input())

print("Enter city names:")
for i in range(cities_third):
    cities_visited.add(input())

print("Cities remaining to be visited:", cities - cities_visited)
```

## Output

```
deadmercury@DESKTOP-QUKDS0U MINGW64 /e/LEARNING/vit_mca (main)
$ C:/Python311/python.exe e:/LEARNING/vit_mca/ITA6017_PYTHON/DA/DA_3/Q5.py
Enter number of cities available:
5
Enter city names:
Bangalore
Vellore
Chennai
Mumbai
Mysore
Enter number of cities visited by first salesman:
2
Enter city names:
Bangalore
Vellore
Enter number of cities visited by second salesman:
1
Enter city names:
Mumbai
Enter number of cities visited by third salesman:
1
Enter city names:
Mysore
Cities remaining to be visited: {'Chennai'}
```

```
deadmercury@DESKTOP-QUKDS0U MINGW64 /e/LEARNING/vit_mca (main)
$ C:/Python311/python.exe e:/LEARNING/vit_mca/ITA6017_PYTHON/DA/DA_3/Q5.py
Enter number of cities available:
15
Enter city names:
Bangalore
Mysore
Chennai
Vellore
Berlin
Delhi
London
Tuticorin
Ranchi
Surat
Itanagar
Patna
Trichy
Mumbai
Pune
```

```
Mumbai
Pune
Enter number of cities visited by first salesman:
3
Enter city names:
Mumbai
Pune
Tuticorin
Enter number of cities visited by second salesman:
4
Enter city names:
London
Delhi
Chennai
Mysore
Enter number of cities visited by third salesman:
1
Enter city names:
Ranchi
Cities remaining to be visited: {'Berlin', 'Trichy', 'Surat', 'Patna', 'Itanagar', 'Bangalore', 'Vellore'}
```

## Question 6

A super market plan to conduct a surprise cum lucky winner contest in a different way. They decide to give a prize for two customers with maximum number of items in common in their shopping list. Write a program to identify common items from the shopping list of two customers. Prepare the common items list as read only.

## Code

```
customerOne = set()
customerTwo = set()

print("Enter number of items in Customer 1 shopping list:")
numOne = int(input())

print("Enter items:")
for j in range(numOne):
    customerOne.add(input())
```

```

print("Enter number of items in Customer 2 shopping list:")
numTwo = int(input())

print("Enter items:")
for j in range(numTwo):
    customerTwo.add(input())

print("First customer's shopping list:", customerOne)
print("Second customer's shopping list:", customerTwo)

commonItems = customerOne.intersection(customerTwo)
commonItemsTuple = tuple(commonItems)

print("Common items in the shopping list:", commonItemsTuple)

```

## Output

```

deadmercury@DESKTOP-QUKDS0U MINGW64 /e/LEARNING/vit_mca (main)
$ C:/Python311/python.exe e:/LEARNING/vit_mca/ITA6017_PYTHON/DA/DA_3/Q6.py
Enter number of items in Customer 1 shopping list:
5
Enter items:
Napkin
Bottle
Maggi
Bag
Pepsi
Enter number of items in Customer 2 shopping list:
6
Enter items:
Pencil Box
Pen
Notebook
Maggi
Napkin
Table Cloth
First customer's shopping list: {'Pepsi', 'Maggi', 'Bottle', 'Napkin', 'Bag'}
Second customer's shopping list: {'Notebook', 'Maggi', 'Pen', 'Table Cloth', 'Napkin', 'Pencil Box'}
Common items in the shopping list: ('Napkin', 'Maggi')

```