

Question 1

Write a Python program which will take N names from the user. Create a dictionary from the N names that will hold First_name, Middle_name and Last_name in separate keys. The inputs will take N at first and then take N names. You can assume that the names will contain less than or equal 3 words and there will be single space between each part of the name.

Sample Input:

```
4
Zubayer Ahmed
Sadia Nur Amin
Mehedi Hasan Shawon
Nafis
```

Sample Output: { "Fname" : ["Zubayer", "Sadia", "Mehedi", "Nafis"] ,
"Mname" : ["Nur", "Hasan"], "Lname" : ["Ahmed", "Amin", "Shawon"]} }

Question 2

One of the important tasks in computational biology is to find short repeating patterns from DNA sequences. These patterns help the biologists to analyze the functionality of different parts in a living being. As a computer science student, you are asked to **write a program that will find all the patterns of a given length from a DNA sequence and their number of presence in that sequence**. This information will help to find the dominant pattern of a given length in the sequence. You will also **find the dominant pattern**. Dominant pattern is the sequence which has maximum occurrence. If there are multiple dominant pattern the program will print "No Dominant Pattern found."

Sample Input:

```
ATTCGATCG
3
```

Sample Output:

```
ATT 1
TTC 1
TCG 2
CGA 1
GAT 1
ATC 1
Dominant Pattern: TCG
```

Sample Input:

ATTCGATCGATC

4

Sample Output:

ATTC 1

TTCG 1

TCGA 2

CGAT 2

GATC 2

ATCG 1

No Dominant Pattern found.

Question 3

Counter Strike Global Offensive is a first person shooting(FPS) game where in a match 2 teams each with 5 members play against each other in different maps. Let's say there are 2 teams, Team A and Team B. For the first half (15 **rounds**) Team A will play as terrorists and Team B members will play as counter terrorists. In the second half(15 **rounds**) Team B will play as terrorists and Team A will play as counter terrorists. The first team to reach 16 rounds will win the match. In order for the terrorist team to win a round they have to plant, defend and explode a bomb at a particular spot or eliminate all the members of the counter terrorist team. In order for the counter terrorist team to win a round, they have to eliminate all the members of the terrorist team or diffuse the bomb that was planted by the terrorist team. Unlike other games, the Most Valuable Player(MVP), can be from any of the two teams regardless of which team wins based on the points calculated from performance of the players. Now you will be given the status of a match between 2 teams as input that will contain performance(Kill, Death, Assist, Bomb Plant, Bomb Defuse, Teammate kill) of all 10 players of the 2 participating teams. Each performance category contains points based on following criteria:

- 2 points for a bomb plant.
- 2 points for a kill.
- 1 point for an assist.
- 3 points for defusing a bomb.
- -1 point for killing a teammate.

Now your task is to find the top 3 players based on the total point. For simplicity, you can consider that no 2 players will have the same point.

Input format:

Team Name	Player Name	Kills	Deaths	Assists	Bomb Plant	Bomb Defuse	Teammate Kills
-----------	-------------	-------	--------	---------	------------	-------------	----------------

Sample Input:

NaVi flamie 23 13 3 2 2 0
Fnatic Xizt 15 20 3 2 0 2
Fnatic KRIMZ 12 23 9 2 2 1
Navi electronic 24 13 5 2 2 0
Navi Zeus 14 22 12 2 2 0
Navi Edward 20 18 8 1 1 1
Fnatic JW 13 24 5 3 0 0
Navi s1mple 26 13 6 3 2 1
Fnatic twist 25 20 4 2 3 2
Fnatic Brollan 14 20 3 2 1 0

Sample Output:

First: s1mple

Second: twist

Third: electronic

Question 4

You will be given a String S which contains only two characters - A and B. Your task is to find some substring pattern P in that string. The pattern P is - a substring will contain an equal number of A and B in each side. For example, in AABAABBAA, we get AB, AABB, BA, BBAA. In this problem, you have to print each pattern P present in the string S along with their frequency. Finally, print the largest pattern P. If there are multiple largest patterns, print the one that came first.

Sample Input

AABAABBAA

Sample Output

AB - 2 times

BA - 2 times

AABB - 1 times

BBAA - 1 times

Longest Pattern - AABB

Question 5

Write a program that reads a long sequence of characters and a word of length 3 with unique characters from the user. Then print the number of times the given word can be written from the characters in the given sequence.

Sample input:

Enter Sequence of characters: that is a cat

given word: hat

Sample output:

Number of times 'hat' can be written from the given sequence: 1

Sample input:

Enter Sequence of characters: Lorem Ipsum has been the industry's standard dummy text ever since the 1500s, when an unknown printer took a galley of type and scrambled it to make a type specimen book.

given word: red

Sample output:

Number of times 'red' can be written: 6

Question 6

As a software engineer, you are hired to arrange cricket players based on their country and type. You have given all the player's information in a tuple called my_tuple. The players are from 3 countries which are Bangladesh, Australia and England. They are 2 types which are "Batters" and "Bowlers"

You are given the following nested tuple where each element in the my_tuple is a tuple consisting of a name and a 4 digit ID.

```
my_tuple = (("Mominul", 1101), ("Mustafiz", 1202), ("Bell", 2101), ("Cook", 2103), ("Smith", 3101), ("Finch", 3102), ("Starc", 3203), ("Imrul", 1103), ("Taijul", 1204))
```

Write a python program that will convert this tuple into the following dictionary using the process described below: **Your code should work for all types of similar tuple if the sample input is changed.**

Output dictionary (You just have to print the resultant dictionary. No need to follow the pattern below)

```
{
  'Bangladesh':
    {
```

```

        'Batters': ('Mominul', 'Imrul'),
        'Bowlers': ('Mustafiz', 'Taijul')
    },
    'England':
    {
        'Batters': ('Bell', 'Cook')
    },
    'Australia':
    {
        'Batters': ('Smith', 'Finch'),
        'Bowlers': 'Starc'
    }
}

```

Here you need to convert the given tuple into the output using the 4 digit ID. The rules are given below:

1==> First digit of the ID represents the country. Here

1a--> 1 means Bangladesh

1b--> 2 means England

1c--> 3 Means Australia

2==> Second digit of the ID represents type. Here

2a--> 1 means Batters

2b--> 2 means Bowlers

3==> If you have multiple players in a category use tuple to store the multiple names as the value of the nested dictionary

4==> If any type is missing from the country, no need to add that type as key in the nested dictionary (e.g: Bowlers is not present as key in the nested dictionary situated in England key)

5==> If your output dictionary's key value pair sequence is not the same as the shown output you should not worry about that. However, the key value pair should be exactly the same

Question 7

You are given a list of courses that are currently available in BRACU that looks like the following:

```
courses = ['CSE110','CSE111','MAT620','CSE520','EEE361','CSE650','MAT510'].
```

All courses till **400 level (499)** are for the **undergraduate** students whereas any courses **above 500** are for the **graduate** or masters students.

Write a python program that creates a dictionary from the above list where the keys will be each department name (CSE,EEE,MAT for the above example) and their values will be a dictionary of two keys -

Undergraduate Courses and Graduate Courses.

Each key will have courses that fall in Undergraduate and Graduate level respectively.

Sample Input - No need to take input

```
['CSE110','CSE111','MAT620','CSE520','EEE361','CSE650','MAT510']
```

Sample Output: (You just have to print the resultant dictionary. No need to follow the pattern below)

```
{
  'CSE':{
    'Undergraduate':['CSE110','CSE111'],
    'Graduate':['CSE520','CSE650']
  },
  'EEE':{
    'Undergraduate':['EEE361'],
    'Graduate':[]
  },
  'MAT':{
    'Undergraduate':[],
    'Graduate':['MAT510','MAT620']
  }
}
```

Question 8

An IP address is divided into 4 sections where each section contains 1-3 digits. An example of an IP address would be 192.168.10.3 . Now IP addresses have 2 parts; One is the network **part** while the other one is the host part. **If the network part is 1, we get a special address from the given IP address where the 1st section will remain as it is and the rest of the 3 sections will become 0. If the network part is 2, we get a special address from the given IP address where the first 2 sections will remain as it is and the rest of the 2 sections will become 0. Same goes for other values of the network part. For simplicity we will consider the network part can be 1 or 2 or 3.**

Lets see an example:

IP address: 192.168.10.30

If network part = 1 , then special address = 192.0.0.0

If network part = 2 , then special address = 192.168.0.0

If network part = 3 , then special address = 192.168.10.0

All IP addresses starting like the special address will fall under the same special network. You will be given the value of the network part and a list of IP addresses as input. Your task will be to

create a dictionary where the keys will be the unique special addresses and the corresponding values will be the list of IP addresses that fall under the same special network.

Input format: Value of network part<Space>List of IP addresses separated by “,”

Sample Input 1:

1 192.168.30.10,168.192.100.30,192.152.55.10,240.169.10.10,168.200.43.90,192.169.100.45

Sample Output 1:

```
{  
"192.0.0.0": ["192.168.30.10", "192.152.55.10", "192.169.100.45"],  
"168.0.0.0": ["168.192.100.30", "168.200.43.90"],  
"240.0.0.0": ["240.169.10.10"]  
}
```

Sample Input 2:

2 192.168.30.10,168.192.100.30,192.168.55.10,240.169.10.10,168.192.43.90,192.169.100.45

Sample Output 2:

```
{  
"192.168.0.0": ["192.168.30.10", "192.168.55.10"],  
"168.192.0.0": ["168.192.100.30", "168.192.43.90"],  
"240.169.0.0": ["240.169.10.10"]  
"192.169.0.0": ["192.169.100.45"]  
}
```