Problem 1

Given the problem statement (below), list the nouns in the upper right box.

In the lower box separate the nouns into two groups.

- Nouns that are likely to be classes
- Nouns that are likely to be attributes

On a distant planet, there is a governing council that is made up of members known as Administrators that are elected from the different regions. The overall leader is the Chief Administrator. Each Administrator may be responsible for multiple departments. The Chief Administrator is a rotating position. This council is frequently changing, as new Administrators are elected, as well as the rotation of the Chief Administrator. Each Administrator's entry tracks their name, region, departments and languages.

Council, Administrators, Chief Administrators, Chief Administrator Biography, Name, Region, Departments, Languages, Administrators Biographies, Classes, Fields … Structure for Application … string types.

Classes:

Council, Chief Administrator Biography, Administrators Biographies.

Attributes:

Administrators, Chief Administrator, Name, Region, Departments, Languages, Fields.

```
class Administrators_Biographies:


    __slots__ = ["Name" , "Region" ,
"Departments" , "Languages"]


    def __init__(Self , Name , Region ,
Departments , Languages):


        Self.Name = Name

        Self.Region = Region

        Self.Departments = Departments

        Self.Languages = Languages
```

Problem 2

Which attributes would be associated with the Administrator class?

Write the code for declaring the Administrator class.

Don't forget to include initializers in the constructor and slots

For now, assume all sequences start empty.

Problem 3

Write a function to add a department to an Administrator.

Use this function and the Administrator constructor from the last slide to create a new Administrator with at least two departments.

```
    administrators_biographies_print_1 =
Administrators_Biographies("Nasqu Baankai." , "Stakins."
, "Interplanetary Affairs and Defense." , "Meinmese and
Vietina.")

    administrators_biographies_print_2 =
Administrators_Biographies("Ittail Xaqe." , "Bhuhleks." ,
"Finance and Transportation and Health Services." ,
"Meinmese and Geulmese.")

    administrators_biographies_print_3 =
Administrators_Biographies("Drincaet Drephral." ,
"Teehors." , "Planetary Affairs and Agriculture." ,
"Meinmese and Ulbiya.")

    administrators_biographies_print_4 =
Administrators_Biographies("Thrilgiens Vraurcaels." ,
"Stadu." , "Education and Justice and Food Management." ,
"Tezniekani and Meinmese.")
```

```
    print("\n")
    print("Administrators Biographies:
")
    print("     Name: " , 
administrators_biographies_print_1.Name)
    print("     Region: " , 
administrators_biographies_print_1.Region
)
    print("     Departments: " , 
administrators_biographies_print_1.Depart
ments)
    print("     Languages: " , 
administrators_biographies_print_1.Langua
ges)
    print("\n")
    print("     Name: " , 
administrators_biographies_print_2.Name)
    print("     Region: " ,
```

Problem 4

Write a print_biography(administrator) function that displays all of the Administrator's information. The information should be formatted to make it easy to read.

Example:

```
Ittail Xaqe
Region: Bhuhleks
Departments:
        Finance
        Transportation
        Health Services
Languages
        Meinmese
        Geulmese
```