#### **Problem Statements**

#### There is a csv file called employee.csv.The file contains employee details with columns: Name, Age, Department, Salary, and Years\_of\_Experience.

1. Write a function department\_with\_highest\_experience() that calculates the average years of experience for each department and returns the department with the highest average experience. (Example Output: 'Engineering', 6.5 )
2. Write a function salary\_by\_age\_group() that groups employees into age ranges (20-30, 31-40, etc.) and calculates the total salary for each age group.(Example Output: {'20-30': 148000, '31-40': 125000, '41-50': 0})
3. Write a function employees\_above\_average\_salary() that returns a list of employee names who are paid above the average salary for their respective department.
4. Write a function give\_raises that increases the salary of all employees with more than 5 years of experience by 10%. The updated data should be saved back into the employees.csv file.

2) A startup is conducting a feedback program of a product, stored in a text file named feedback.txt, where each line represents one piece of feedback. The company wants to analyze the feedback to understand customer sentiment around certain aspects, such as "service," "price," and "quality." Each feedback line could express either positive or negative sentiment. Positive feedback often contains keywords such as "excellent," "great," "satisfied," and "happy," while negative feedback includes words like "poor," "bad," "disappointed," and "unsatisfied." Write a Python program to read feedback.txt, analyze the following.

* 1. Count keyword occurrences: Count how often each specified keyword appears in the feedback.
  2. Separate good and bad feedback: Save positive feedback lines in a file named good\_feedback.txt and negative feedback lines in bad\_feedback.txt.
  3. Summarize counts: Save the keyword counts and the overall counts of good and bad feedback lines in keyword\_counts.csv, without overwriting any existing data in this file.

Avoid overwriting- Ensure that no previous data is lost or overwritten in good\_feedback.txt and bad\_feedback.txt; if these files exist, append new data to them.

3) A restaurant chain wants to calculate the final cost of a meal, taking into account various discounts, service charges, and taxes. Write a program that has 3 functions that call each other to get the final price. Each step involves multiple conditions based on default parameters:

* 1. apply\_discount: This function reduces the base price by a default discount rate (e.g., 10%). Additionally, if the customer is a member, it applies an additional loyalty discount (default 5%). If it’s a special promotion day, it applies a further discount (default 3%).
  2. add\_service\_charge: Calls apply\_discount, then adds a service charge based on the type of meal (dine-in or takeout) with default rates. It adds a flat surcharge for dine-in orders, whereas for takeout, it calculates a percentage of the discounted price.
  3. calculate\_final\_price: Calls add\_service\_charge and applies a tax. For high-value orders (default threshold $100), an additional luxury tax (default 2%) is applied.

In the main program, call calculate\_final\_price with parameters to test different scenarios. Make sure to use default parameters wherever required.

4) You have a folder with 10 students output files and a main key output file. Each student output file and the key output file contain 10 lines of text. Your task is to compare the content of each student output file with the key file line by line and assign marks based on the correctness of each line. Each line is worth 2 marks, so the maximum score for a student is 20 marks. (files required are given.)