Create following programs:

Create a program that will read content of a file employees.txt and put employees in a Map company that will keep information on department as key and will have list of Employees (class should have first name and last name) as a value. Finally create as many files as there are keys in Map and save employees to corresponding file.

Create a program that will read content of file employees.txt store employees in an Array List, where class Employee should have first name, last name, salary and department as attributes. Once you process all lines, find employee with highest and lowest salary using threads. You can complete this task by using two separate, one for highest and one for lowest. classes implementing Runnable interface. For highest and lowest salary use at least two threads. When threads are completed (make sure you call join method), output result to a file. For synchronisation of data use static class with lock method.

Create a program that will read content of a file employees.txt and save employees to ArrayList, where class Employee should have first name, last name, salary and department as attributes. After read content of a file managers.txt and save mangers to ArrayList, where class Manager should have department name, manager name and allocated budget. Once everything is processed, calculate following:

Total of salaries for employees for each department and remaining in the budget for each manager. Manager budget - (total salaries in that department). Print result to a console.

Top 10 salaries in each department. Print results in a file per department.