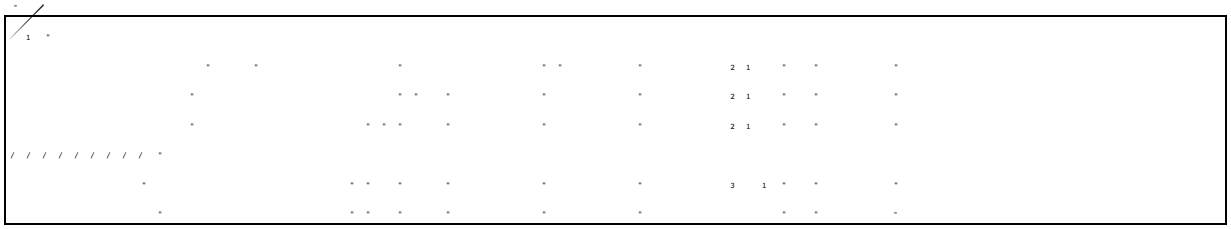
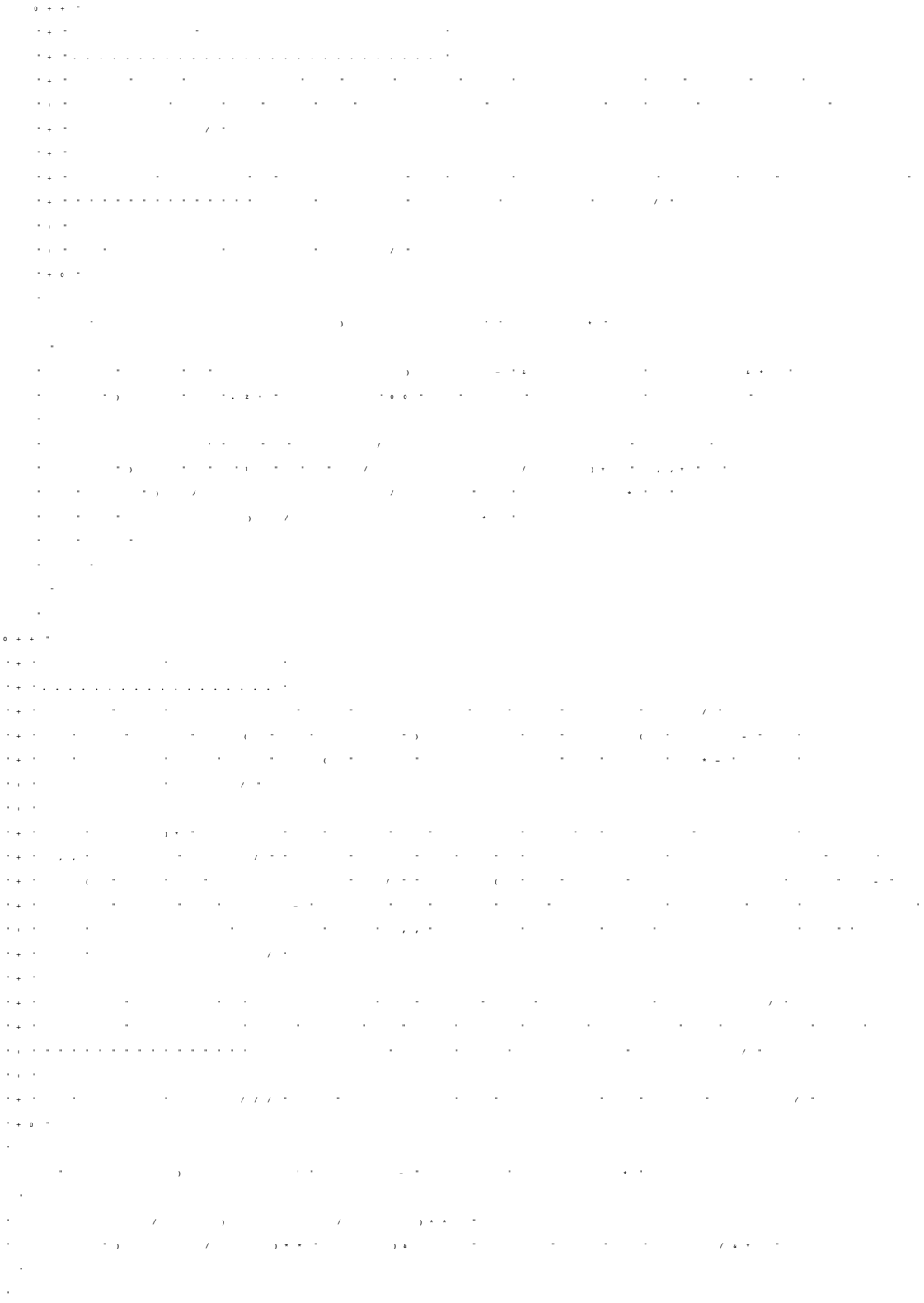
# Crash Course: Records and Files

Your task is to design a data structure to keep track of the academic departments on campus. There are an unbounded numbers of departments, and each department has a name and a variable number of instructors and course numbers. For each instructor, you need to keep a name, a salary, whether or not the professor has tenure, and the instructor's level. An instructor may be one of four different levels: a lecturer, an assistant professor, and associate professor, or a full professor. You should represent a level with an enumerated type. For a course, you should record the name, course number, number of units, and which quarters the course is offered. There should be one encompassing data structure called **`university`** that represents the database for an entire school. We track the name of the school and the full set of all departments.

a) First, specify all of your data structures.
b) Write a function that given one department as a parameter will return the total number of course units that department teaches **`Winter`** quarter.
c) Write a function that given a **`university`** as a parameter will promote all the CS lecturers. This means that you must first find the department named **`"Computer Science"`** in the database and go on to promote their lecturers (cause we're so abused.) Promoting a lecturer means giving him tenure, bumping his level to Assistant Professor and doubling his salary. You should write helper functions as needed.
d) Write a function **`InitDepartment`** that accepts a reference to a department and initializes the data structure with values read in from two files. The function should take three **`string`** parameters: the name of the department, the name of the file containing the instructor data, and the name of the file containing the course data. Assume the files are properly formatted so that no error checking is required.

**`CSInstructors.data`**



**`CSCourses.data`**

## Data Structures