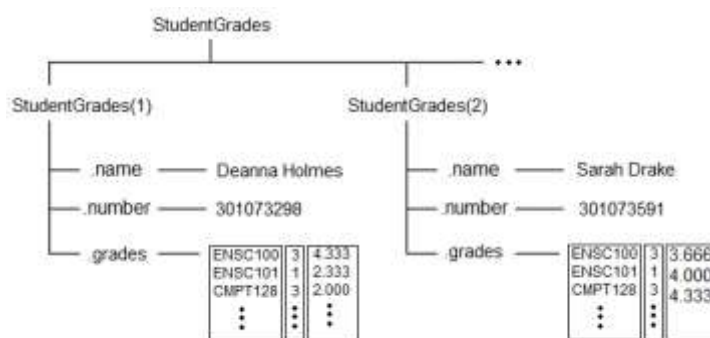


Lab 4

- Write a function that takes a matrix as its input. The entries of the matrix are non-negative integers of type `double`. The function returns the variable type (e.g., `uint8`, `uint16`, `uint32`, and `uint64`) of the unsigned integer class to which the matrix can be accurately converted. For example, if the largest integer in the matrix is 14, the function returns `uint8`. If no such class exists, the string `'NONE'` is returned.
- Write a script that generates a cell array of dimensions 31-by-3. The rows of the array correspond to the days of October, 2015. The three elements of each row must be set as follows:
 - The first element refers to the string `'October'` (uppercase `'O'`).
 - The second element refers to a scalar of type `double` that equals the date (1 through 31).
 - The third element refers to the three-letter abbreviation of the day chosen from this list: `'Mon'`, `'Tue'`, `'Wed'`, `'Thu'`, `'Fri'`, `'Sat'`, and `'Sun'`.

The eleventh element of the cell array should be: `'October' [11] 'Sun'`.

- Write a function that takes a string representing an integer between 1 and 20 inclusive using Roman numerals and returns the Arabic equivalent as a number of type `uint8`. If the input is illegal or its value is larger than 20, the function returns 0 instead. Allow only three consecutive symbols can be the same in your function. For example, `IIII` or `VIII` are illegal.
- Write a script that reads a data structure `StudentGrades` from `"04Student_grades.mat"`. The structure is organized as the follows:



The fields are:

- `name` – a string specifying the student's name.
- `number` – a scalar specifying the student's number.
- `grades` – a 1-by-3 cell array where:
 - The 1st cell is a 9-by-7 char matrix specifying the course names.
 - The 2nd cell is a 9-by-1 vector with the corresponding credits for each course

of the 1st cell.

- The 3rd cell is a 9-by-1 vector with the corresponding grade that the student earned for each course of the 1st cell. These are in the form of a 4.333 grade scale (e.g., 4.333 is an A+, 4.000 is an A, 3.666 is an A-, 3.333 is a B+, etc.).

The script should calculate each student's cumulative GPA (CGPA) for a set of N courses, i.e.,

$$\text{CGPA} = \frac{\sum_{i=1}^N c_i g_i}{\sum_{i=1}^N c_i},$$

where g_i and c_i are the grade number of credits for the i th course. The script should write the students' names, numbers, and CGPAs into an Excel XLSX file. Include the header in the file.

5. Write a script to generate a table of the logarithm values of base ten, i.e., $y = \log_{10}x$. The x ranges between 1 and 10 in steps of 0.1. The results should be arranged in a table as shown below. The table should include a title in the first row, and row and column headings. The script should store the table into an Excel XLSX file.

(Title)										
	.0	.1	.2	.3	.4	.5	.6	.7	.8	.9
1.0	0.000	0.041	0.079	0.114						
2.0										
3.0										
4.0										
5.0										
6.0										
7.0										
8.0										
9.0										
10.0										

6. Download the file "04Stocks.txt" from the course website. Write a script to open the file and read all the data to memory. Your script should sort the data by the 3rd column in ascending order. Your script should then save sorted data to a new file "Stocks_sorted.txt".