Математические пакеты

Домашнее задание №3 (Octave)

Дедлайн до 25.09 23:59

Не используйте циклы в заданиях, если явно не сказано обратного!

(1) (2 балла) Функция `all2dets (X)`. Дана матрица `X` из двух столбцов. Создать квадратную матрицу, размером в количество строчек исходной. Элемент в `i`-ой строке `j`-ом столбце должен быть равен определителю матрицы `[X(i,:); X(j,:)]`, т.е. матрицы 2×2, составленной из `i`-ой и `j`-ой строк исходной матрицы. Напомню, что $\det\begin{pmatrix} a & b \\ c & d \end{pmatrix} = ad - bc$

(2) (4 балла) Функция `all_lines_intersections (fname)`. На вход подаётся переменная-строка `fname` имя txt-файла (без расширения!), в котором записана матрица из трех столбцов (элементы разделены табуляцией), описывающая набор прямых. Если кратко сформулировать условие, то необходимо пересечь каждую прямую с каждой. Подробней так: каждое пересечение прямой из строки `i` с прямой из строки `j` дает две координаты пересечения `x` и `y`. Функция должна вернуть две матрицы `x` и `y`. Первая содержит `x` координату этого пересечения в строчке `i` и столбце `j`, вторая содержит, соответственно, `y` координату. Не обрабатывайте отдельно случаи параллельных прямых, пусть для параллельных прямых получается произвольный ответ, скорее всего, у вас это будет NaN из-за деления 0 на 0. Напомню формулу пересечения прямых $a_1x + b_1y + c_1 = 0$ и $a_2x + b2_y + c_2 = 0$:

$$\Delta = \det\begin{pmatrix} a_1 & b_2 \\ a_2 & b_2 \end{pmatrix},$$

$$\Delta_x = \det\begin{pmatrix} -c_1 & b_2 \\ -c_2 & b_2 \end{pmatrix},$$

$$\Delta_y = \det\begin{pmatrix} a_1 & -c_1 \\ a_2 & -c_2 \end{pmatrix},$$

$$x = \Delta_x/\Delta,\ y = \Delta_y/\Delta.$$

Соответственно, пользуйтесь задачей (1) для решения.

(3) Функция `arab2rome (a)`:

(a) (1 балл) На вход подаётся вектор, состоящий из натуральных чисел от 1 до 10. Необходимо вернуть список той же длины, элементы которого содержат в себе римские цифры. e.g.: вектор `a = [1, 1, 2, 3, 5, 8]` превращается в список `r = {"I", "I", "II", "III", "V", "VII"}`.

(b) (2 балла) Модифицируйте функцию. Теперь на вход поступает вектор с различными натуральными числами. Верните список, в котором числа от 1 до 10 заменены римскими аналогами, а вместо остальных чисел будет символ доллара. e.g.: вектор `a = [1, 1, 2, 3, 5, 8, 13]` превращается в список `r = {"I", "I", "II", "III", "V", "VII", "$"}`.

(4) (3 балла) Функция `palindrome (str)`. На вход поступает строка `str`, которую надо проверить, является ли она палиндромом. В случае положительного или отрицательного ответа выведете соответствующее сообщение на экран. e.g.:

```
>> palindrome ('"Not New York", - Roy went on')
>> This string is a palindrome!
>>
>> palindrome ("Hello, world!")
>> This string is not a palindrome!
```

(5) (5 баллов) Функция `jabberwocky ()`. В файле `Jabberwocky.txt` находится стихотворение «Бармаглот». На вход функции подаются строки (это могут быть как буквы, так и слова), количество строк неограниченно. Вам нужно определить количество вхождений каждой строки в стихотворении и номер строки, в которой она встречается впервые. Верните результат в виде файла `jabberwocky-analysis.txt`. Вы можете воспользоваться одним циклом по входным строкам, но не пользуйтесь циклом при выполнении анализа. *Примечание: в начале файла имеются 3 строчки, которые нам не нужны в анализе. Попробуйте найти функцию, которая пропускает эти лишние строки.*