Cells and Structs

Exercises

randomsentencecell

- Write a function **randomsentencecell.m** that returns a random sentence using the following names, verbs, and nouns:
  - names: Harry, Xavier, Sue
  - verbs: loves, eats
  - nouns: baseball, rocks, sushi
- In your function, you must construct a single cell array variable `parts` that contains the names, verbs, and nouns. (Hint: nested cell arrays)

```
>> randomsentence
ans =
Harry loves rocks.
>> randomsentence
ans =
Sue loves sushi.
```


randomsentencestruct

- Write a function randomsentencestruct.m that does the same thing as the randomesentencecell function described above. The variable `parts` should now be a single struct containing fields "names", "verbs", and "nouns", where each of the values of these fields is a cell array of strings.

Letter grades and gpa

- Consider the following grade conversion table:

F	D-	D	D+	C-	C	C+	B-	B	B+	A-	A	A+
0	60	63	67	70	73	77	80	83	87	90	93	97

- Write a function getlettergrade.m that takes a percent grade and returns the letter grade. Negative grades get an F, and grades over 100 get A+. Try solving this problem without for loops.

glucoselevels

- You are given a cell of strings containing gender and blood glucose information of some patients. Gender is given as 'f','female','m' or 'male'. Glucose levels are given as mg/dL, but some patients are missing the glucose level information.
- Write a function to convert the cell to **a matrix of double**. Use 0 to represent female, 1 to represent male. Missing data values should be returned as NaN.
- Your function should return the average glucose levels of the requested gender (remove any NaN before you calculate the average).
- glucoselevels({'f' '100'; 'm' '200'; 'f' '120'; 'female' ''; 'male' '180'}, 'f')