CS106 W21 - Assignment 06

Due: Friday, March 12, 11:59 PM

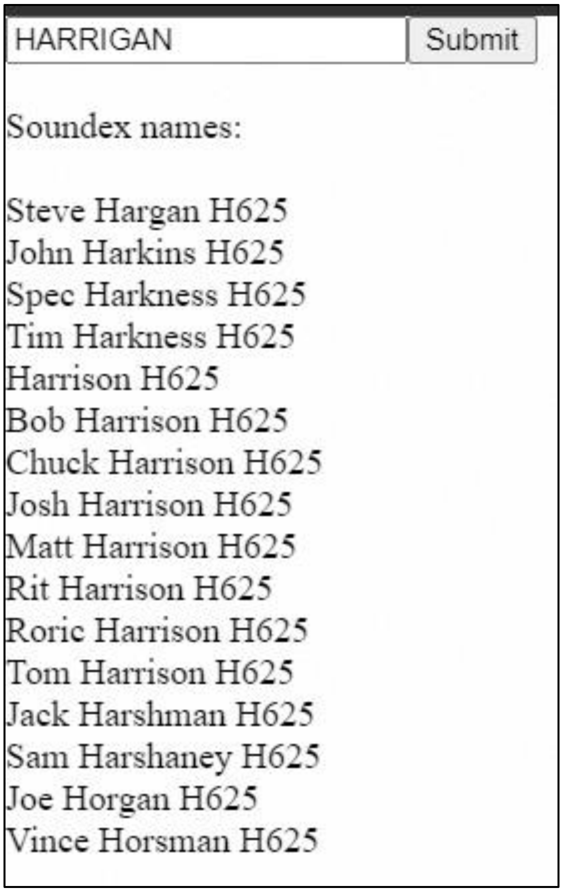
Assignment 6 is graded out of 26 marks.

Soundex Algorithm

The Soundex algorithm allows us to find last names that sound like other last names. For example, Smith sounds like Smythe. This is done by determining a 4 digit soundex code for each name, and if those soundex codes are the same then we say the names sound alike. For example, Harrigan becomes “H625” and Hershenson becomes “H625” so we say Harrigan sounds like Hershenson. Smith and Smythe are both “S530” so we say they sound alike. If you google the internet you will see various algorithms for soundex. In this assignment you will write a function to implement the soundex algorithm and then use that function to find names that sound like other names.

The version of the soundex algorithm we will use is as follows:

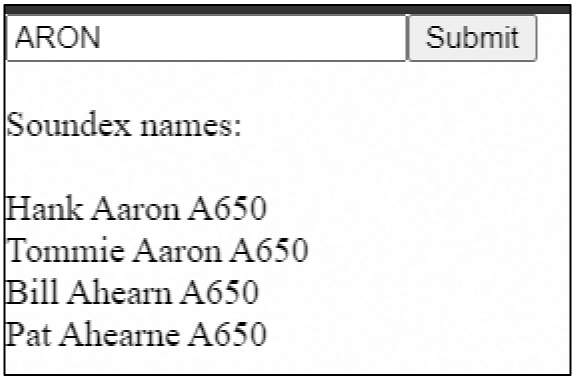
1) Retain the first letter. If the name is “Jones” then retain “J”. If the name is “Trump” retain the “T”.
2) Replace each subsequent letter with a number using the following:
    - a, e, i, o, u, y, h, w → 0
    - b, f, p, v → 1
    - c, g, j, k, q, s, x, z → 2
    - d, t → 3
    - l → 4
    - m, n → 5
    - r → 6
3) At this time we have a long soundex code including zeroes. For example, Harrigan at this point is the String: “H0660205”. At this point the length of the original name “Harrigan” and the soundex code “H0660205” are always the same length (length of 8 in this example name). Now, for any adjacent pairs, you delete the second occurrence. In this example we remove the second “6” and thus “H0660205” becomes “H060205”.
4) You then remove the zeroes. “H060205” becomes “H625”.
5) If the soundex code has a length less than 4, then you pad it with zeroes on the right. For example, the name “Munn” becomes “M5” after the previous step, and it is then padded with zeroes to the right to have a total length of 4, and thus it becomes “M500”.
6) If the soundex code has a length greater than 4, then you truncate it. For example, the name “Delaportilla” becomes “D41634” after step (4) above, then you truncate it to a length of 4, and thus it becomes “D416”.
7) The soundex code of length four is returned by the function soundex().



HARRIGAN

```
Soundex names:

Steve Hargan H625
John Harkins H625
Spec Harkness H625
Tim Harkness H625
Harrison H625
Bob Harrison H625
Chuck Harrison H625
Josh Harrison H625
Matt Harrison H625
Rit Harrison H625
Roric Harrison H625
Tom Harrison H625
Jack Harshman H625
Sam Harshaney H625
Joe Horgan H625
Vince Horsman H625
```



ARON

```
Soundex names:

Hank Aaron A650
Tommie Aaron A650
Bill Ahearn A650
Pat Ahearne A650
```

Your task is to write a sketch to mimic what you see in this video:
https://vault.cs.uwaterloo.ca/s/GpQpczLgQPRTHdK

There is starter code at: https://openprocessing.org/sketch/1123339

The starter code does the following:

- Loads the baseball master table “Master.csv”. It contains the first and last name for each player (as well as many other columns of data about each player).
- Allows the user to type in a name and press a “submit” button.
- When the user types a name and presses “submit”, it runs a function named “findSoundexNames”.

Write a sketch to do the following to mimic what you see in the above video:

a) [15 marks] Your soundex function works as follows:
    - [1 mark] It has a function header that is “soundex(s)”. (this is shown in the starter code)
    - [14 marks] The implementation of “soundex(s)” follows the seven steps above.

b) [5 marks] Your function “findSoundexNames” finds all matching players who have a last name that “sounds like” the name the user typed in.
    a. [1 mark] It must determine the soundex code of the name that was typed in (by calling the soundex function).
    b. It must loop through every row in the “Master.csv” table and find all last names that match the typed in name.

c) [1 mark] For each player that matches, you display the player’s first name, last name, and the soundex code. The matching names are displayed on the html page. Do not use a canvas (ie. Use noCanvas()).

[5 marks] Coding Style and Efficiency

Follow the course coding style for whitespace and comments. Consult the **“Code Style Guide”** on LEARN. For example:

1) [0.5] Include your name on the first line of code and your student ID number on the second line of code.
2) [0.5] Leave the third line blank.
3) [0.5] Comment your code appropriately. Avoid superfluous comments.
4) [0.5] Correctly and consistently indent your code blocks.
5) [0.5] Use correct inline spacing for variable declaration and assignment.
6) [0.5] Use good line spacing to chunk sections of your code.
7) [0.5] There are no variables that are declared or assigned, but not used.
8) [0.5] There are no unnecessary variables that are duplicates of other variables.
9) [0.5] There is no unnecessarily repeating the same code in multiple places.
10) [0.5] Semicolons were used appropriately (i.e. at the end of most lines).

Restrictions

- You may not use any functions or statements not covered in lecture or labs.
  This includes, but is not limited to:
  - No translate(), rotate(), or scale() functions.

Submitting

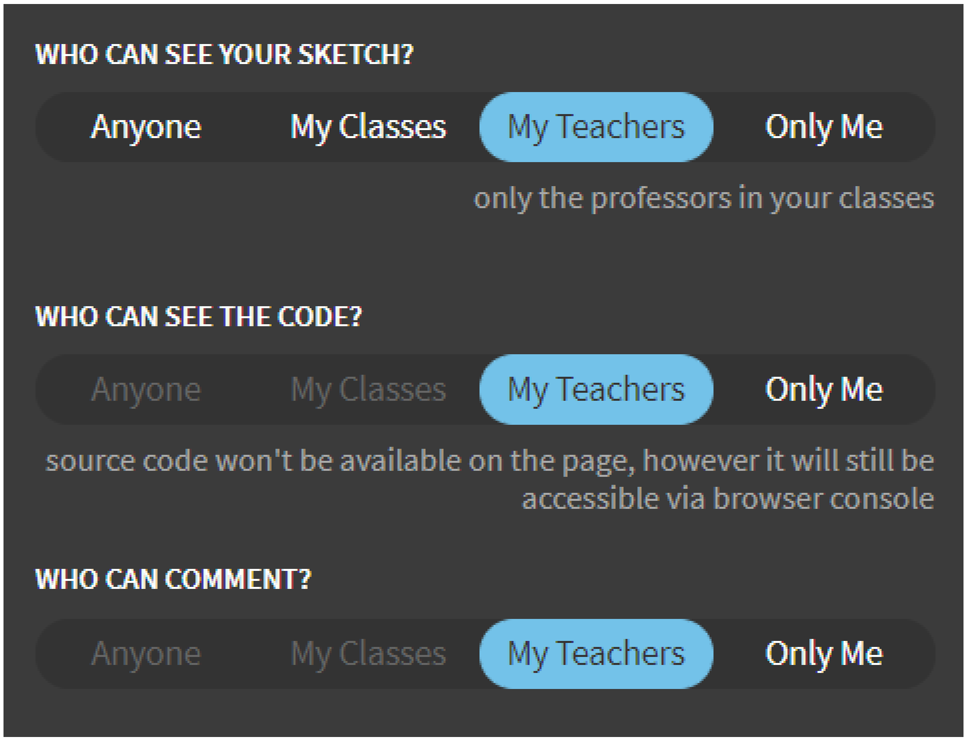
Use the template file in Word “CS106 Assignment Template” in LEARN to create your Assignment 06 submission.

Then convert your Word file to pdf. Please ensure that your URLs are hot links. The TAs need to be able to click on each link in your pdf and go directly to your sketch.

So for example, don’t have a link like this: https://openprocessing.org/sketch/1050954

but rather have that link as a hot link as follows: https://openprocessing.org/sketch/1050954

Ensure that each URL you submit has its settings so that the access is as follows:



WHO CAN SEE YOUR SKETCH?

Anyone | My Classes | [✓ My Teachers] | Only Me

only the professors in your classes

WHO CAN SEE THE CODE?

Anyone | My Classes | [✓ My Teachers] | Only Me

source code won't be available on the page, however it will still be accessible via browser console

WHO CAN COMMENT?

Anyone | My Classes | [✓ My Teachers] | Only Me

Submit that pdf file to the Assignment 06 dropbox on LEARN.

An example of how to submit a Lab is shown in the following video: https://vault.cs.uwaterloo.ca/s/9Xx7AGsewaea773

It is your responsibility to submit to the correct dropbox with the correct file before the deadline. Otherwise you may receive a mark of 0.

Academic Integrity

All assignments in CS106 are done individually. Group work and sharing of code is not allowed.

Detecting Plagiarism:

- We monitor Reddit, File Trading Sites, past year CS106 assignments, etc.
- Measure Of Software Similarity (MOSS)
  - automatic system for determining the similarity of code

Discipline

- Discipline (Policy 71)
  - https://uwaterloo.ca/secretariat-general-counsel/policies-procedures-guidelines/policy-71