Homework #4, ENGR 100-430, W24. Due **Fri. Mar. 8, 5PM**

Notes

- This is an individual assignment, not a group project. Refer to the course syllabus for the collaboration policies.
- The purpose of this assignment is to help you better understand the `Julia` commands that are particularly useful for music signal processing.
- Some of these problems you can solve just by typing the commands into `Julia`. For example, if you are asked what the output of the command `sum(ones(6,5))` is, you can type that in and find that the answer is `30`. But you need to understand what is happening well enough that in the future you could answer such questions without using `Julia` itself.
- To help you develop this skill, some questions will be expressed in terms of *variables*, rather than specific numerical values, and you must express your answer in terms of those variables. For example, if you were asked what the output of the command `sum(ones(m,n))` is, you assume that `m` and `n` are scalar variables that were previously defined, and the correct answer is "mn," *i.e.*, the product of the values of those two variables. For such problems if you do not know the answer right away, you can experiment with specific values of `m` and `n` to see what is happening, *e.g.*, by typing something like: `m=4; n=5; sum(ones(m,n))`. But after you understand it you must report your final answer in terms of the variables such as `m` and `n`.

---

1. [0] What is the size (# of rows and columns) of the array `y` produced by the following `Julia` statements?
```
x = 1:10; y = reshape(x, 5, 2)
```

---

2. [2] What are all of the pairs of values of `m` and `n` for which the following `Julia` commands will execute without errors? `x = 1:12; y = reshape(x, m, n)`

---

3. [0]
   (a) What is the size of the array `p` produced by the following statements?
```
a = 10:5:50; b = 4:-1:1; p = a * b'
```
   Hint: you can use the `size` function to find a variable's size.
   (By the way, the mathematical term for a column vector times a row vector is a outer product.)
   (b) What does `p[2,:]` return?

---

4. [3]
   (a) What is the size of the array `c` produced by the following statements, assuming that `n` and `m` are even?
```
a = 0:2:n; b = 1:2:m; c = a * b'
```
   (b) What is the size of `c[:,3]`, assuming that `m` ≥ 5.

---

5. [0] What is the size and value of the variable `d` produced by the following statements?
```
a = ones(9); b = 1:9; d = a' * b
```
(By the way, the mathematical term for a row vector times a column vector is an inner product.)

---

6. [2] What is the size and value of the variable `f` produced by the following statements?
```
a = ones(m); b = 3*a; f = a' * b
```

---

7. [4] The following `Julia` code makes a sinusoidal signal array and reshapes it into one long signal—a song.
```
S = 8192; x = 0.9 * cos.(2pi/S * (1:2000) * (150:50:350)')
y = vec(x); sound(y, S)
```
   (a) How many notes are in this song and what are the frequencies of each note?
   (b) What is the duration of this (short) song?

8. [4] Drawing inspiration from the previous problem, write down a sequence of at most 6 `Julia` statements (with no semicolons!) that synthesizes the first 8 notes of the chorus of the Pink Floyd song "Another Brick in the Wall," *i.e.*, the part with lyrics "we don't need no eduction." Hint. This part of the song is in the key of D minor and the notes are D E F E D E F E. Each note in your song should be 0.25 seconds long. (So the rhythm will not quite match the original.) The last of your statements will be `sound(y, S)` . Try your code in `Julia` to check it is correct.

9. [2] The following `Julia` statements generate samples of a sawtooth wave signal and then plays the sound:
```
using Sound; x = 0.1 * (mod.(0:3999,20) .- 10); sound(x, 8192)
```
Assuming that the sampling rate is $S = 8192$ $\frac{\text{Sample}}{\text{Second}}$, determine the fundamental frequency of this signal.
Hint. Try `plot(x[1:90], marker=:circle)`

10. [2] Assuming that the sampling rate is $S = 8192$ $\frac{\text{Sample}}{\text{Second}}$, and assuming that `n` $\gg$ `m` , determine the fundamental frequency of the (periodic) sampled signal generated by the following `Julia` statement:
```
x = mod.(1:n,m) .- m/2
```

11. [2] Consider the following `Julia` statements.

```
x = 0.9 * cos.(2pi * 151 * (0:1999)/8192)
Y = cos.(2pi * (0:1999) * (100:50:350)'/8192)
corr = Y' * x
index = argmax(abs.(corr)) # or findmax(abs, corr)[2]
```

   (a) [0] What is the size of the (correlation) array `corr` ?

   (b) [0] What is the value of the variable `index` ?

   (c) From a signal processing perspective, why is the 2nd element of `corr` the largest?