Memory Recall Task  
I: Project Description

II: Requested changes  
  
**I: PROJECT DESCRIPTION.**   
  
  
This is a word recall study. We are testing participant’s ability to recall lists of words. Additionally, we are looking at whether types of background noise, and memory strategy impact memory performance. There are several types of memory tasks included on the project: Complex, Running, Serial Recall, Delayed Serial Recall, and Free Recall.  
  
  
Additionally, there are 3 sound conditions that occur during these tasks:   
a. repeated sound.

b. changing sound.

c. silent  
  
For each of the tasks, the participants are shown 7 words 1 at a time. They are then asked to recall them (the type of recall is different for each task). For each task, this occurs for 3 practice rounds, and then 9 test rounds.

During each round of each task there will be will either be silent, repeated sounds (steady state), or changing sounds (changing state). This is set to randomize so that it each would happen an equal amount of times, and in random order. The sounds play during the presentation of the words, and up to the moment where they have to do the recall portion of the task.

Here is an overview of the sequence of this task:   
  
The way the task goes is:   
  
1. Sound check.

2. Description of the task (instructions)

3. Strategy selection.

4. Sound Warning (participant is informed that there will be background noises and to try to disregard them and remember the words.

5. Practice Trial (3 rounds)

(*note: in random order, each of the three sound conditions will occur during practice)*

End Practice

6. Strategy change/keep

7. Test Trial (9 trials)  
  
  
**II: REQUESTED CHANGES:**

**a.** I will not be using complex, running, or non-delayed serial recall. I address this first so that you don’t end up making unnecessary fixes to the tasks which I won’t be using and which you will remove anyway.   
  
 As of now, the code allows for one task to be run at a time depending on the input for ‘task’ during the first window. If nothing is entered for ‘task’ all of the tasks are presented in random order.

I will be asking you to add a new task called item recognition. I will explain that task in greater detail below.   
  
The final code only needs to run through the following three tasks, in the following order:

1. Free recall

2. Item Recognition

3. Delayed Serial Recall.   
  
When nothing is entered for ‘task’ I’d like the code to run those three in the order presented.

**b.** I want the strategy “checklist” removed from the options of strategies. This will need to be removed wherever the code allows for a selection/re-selection of strategies.

**c.** In each round of the memory tasks (3 practice, 9 test rounds), 7 words are

randomly selected from a larger list of words. I want to change the initial, larger pool of words from how it is currently set up….  
  
I want there to be *three* separate initial lists of 16 words. You can use random words for now, and I will replace them with our words later.

This is how I’d like the tasks to function.   
First, there will be the initial word lists:   
  
Wordlist A (16 words)  
Wordlist B( 16 words)  
Wordlist C (16 words)  
  
Each time a participant does this study, each of the 3 tasks (1. Free recall , 2. Item Recognition 3. Delayed Serial Recall. ) will be randomly assigned one of those 16-word, wordlists. Then, during each round of that task (practice and test) of those tasks, 7 random words are pulled from the word pool for use in that task.

**d.** After the 3 practice rounds of each task, there are currently 9 trial runs, I want this to be bumped up to 18 rounds for all tasks.

*This will mean that there are 6 of each background noise conditions (silence, steady state, changing state) per task.*

**e.** As of now there is a list of files for the background noises: 18 CS (changing state), and 18 SS (steady state) files. The repeated or changing sound conditions randomly shuffle through these files. Instead, I want there to only be 4 of each kind of files. (CS1, CS2…..CS5), . (SS1, SS2…..SS5).

**f.** Item Recall Task: This is the new task I’m asking you to develop. If anything in this task description unclear, I am happy to explain it more thoroughly over zoom or on the phone.

Task description:   
  
- Like the other tasks, the item recall task will also have 3 practice rounds followed by 18 rounds. The background sound conditions will have the same sequence (6 of each condition across the 18 rounds, randomly shuffled)  
  
 -Like the other tasks, each round of the item recognition task will first show the participant 7 words (randomly picked from the pool of 16)

- Next, for the recall portion of this task, the participant will be shown a word from the 16-word list and ask if it was one of the 7 just showed in this round. The participant will click 1 for yes, and 2 for no.  
  
- For half of the rounds, the word they are shown may be a ‘**lure’** word. In this case the word is not one of the 7 they were just shown, and ‘no’ is the correct answer. For the other half of the rounds, the word will be a ‘**probe’** from one of the 7 word’s they were just shown, and ‘yes’ will be the correct answer. Half the rounds should be lure rounds, while half the rounds will be probe rounds.   
  
- One more important thing is that the lure rounds vs. probe rounds should be split *within* each background noise condition:Like the other tasks, Silence, Changing State, and Steady State background noise should have 6 rounds each. For each noise condition, there should be 3 probe rounds and 3 lure rounds.

Output csv data from this task should mirror the other tasks, but in the column ‘condition’ this task should output ‘IR’ for each round, the 7 presented words for each round will still fall into the item1…item2 columns.   
  
But there should be the following new columns  
ir\_word: will display which word was presented to them  
ir\_condition: ‘probe’ or ‘lure’

Ir\_answer: will show if they answered yes or no  
ir\_score: will be a 1 if they were correct and a 0 if they were incorrect