

# Savio Concessio - ENGR101 Project 1 Pseudo Code

## CORE

1. SET “n\_samples” as chosen sample rate multiplied by duration (seconds) sound should play for
2. Declare and initialise array to hold air pressure at an instance with a size of “n\_samples”
3. FOR each element in the array, starting from the first one
  - a. Set array element to value of air pressure of sound wave formula at iteration
4. END FOR
5. Call MakeWavFromInt function to convert array to a sound file
6. Clear “waveform” from memory
7. RETURN 0

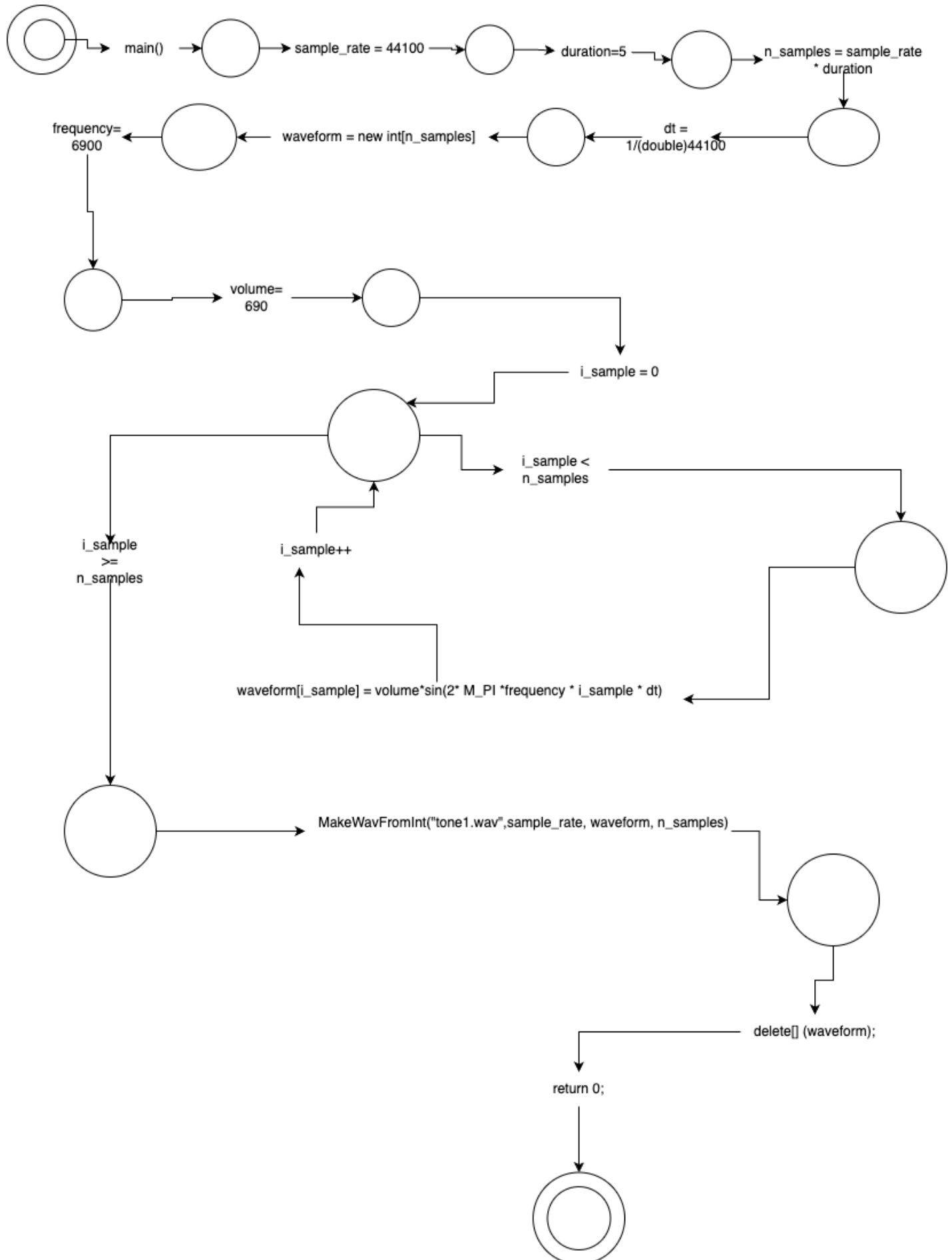
## COMPLETION

1. Set “freq1” to first frequency chosen for siren
2. Set “freq2” to second frequency chosen for siren
3. Set “totalSamps” as chosen sample rate multiplied by duration (secs) sound should play for
4. Declare and initialise array to hold air pressure at an instance with a size of “totalSamps”
5. FOR number of notes played
  - a. Declare local variable “freqNow”
  - b. If an odd iteration - set “freqNow” to “freq1”
  - c. Else set “freqNow” to “freq2”
  - d. FOR one quarter of sample rate chosen, starting from 0
    - i. Set “calculated\_value” to value of air pressure formula at iteration
    - ii. Set array element at j shifted by number of tones already played to “calculated\_value”
6. END FOR
7. Call MakeWavFromInt function to convert array to a sound file
8. Clear “waveform” from memory
9. RETURN 0

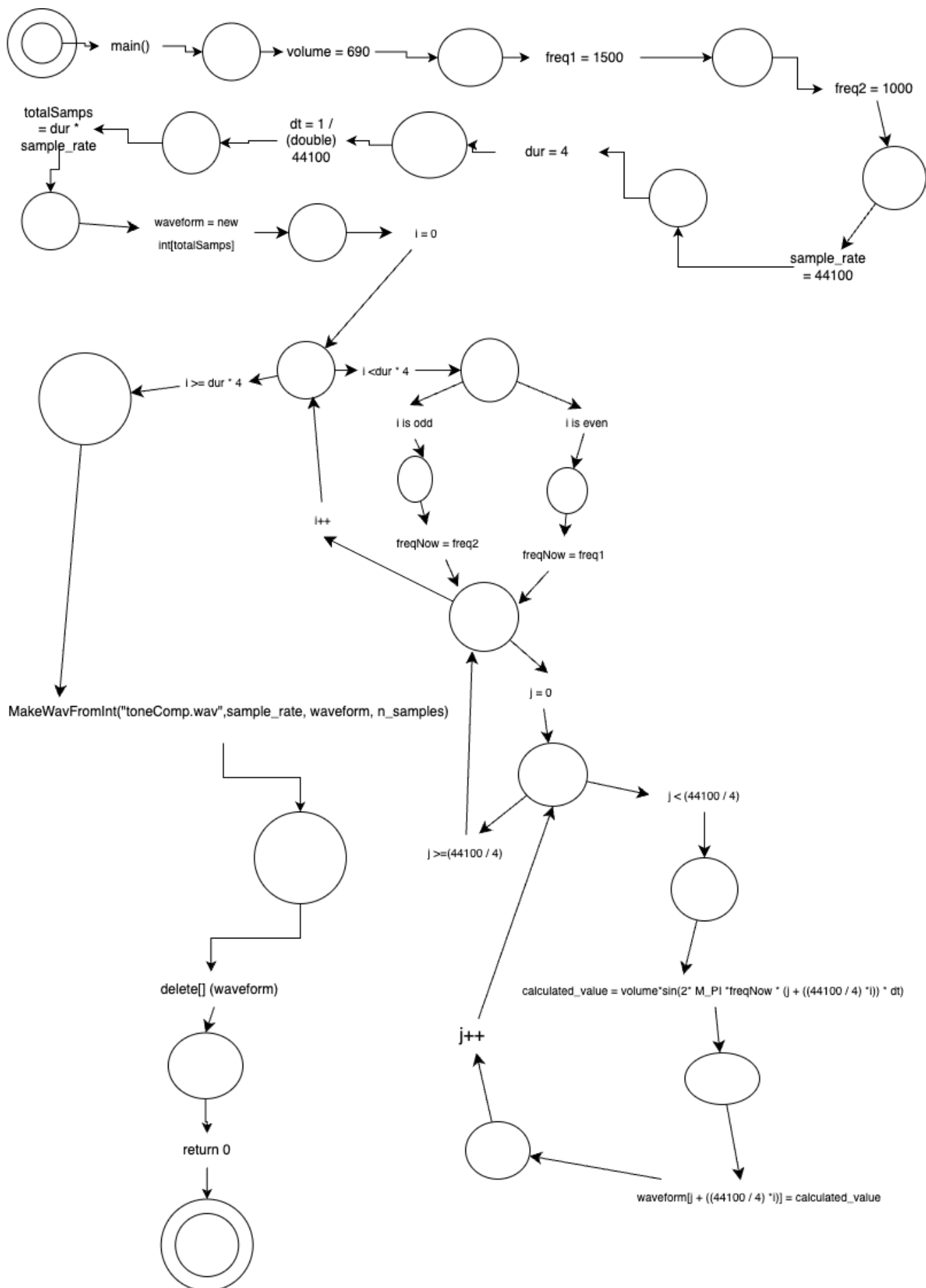
## CHALLENGE

1. Declare vector “waveform” to hold air pressure at all samples of a song
2. Declare a stream to read text from the selected file
3. If file did not load/contained an issue
  - a. RETURN 1
4. Declare variable “i” as index to calculate air pressure at
5. WHILE file still has values to read
  - a. Read and save next input from file
  - b. Convert file input to number and set to “freq”
  - c. FOR number of samples chosen for one note, starting from 0
    - i. Declare variable “thisVolume”
    - ii. If iteration of note is below the chosen peak point of the note
      1. Set “thisVolume” to a positive gradient calculated from the iteration of this note.
    - iii. Else set “thisVolume” to a negative gradient calculated from the iteration of this note.
    - iv. Set “val” to value of air pressure formula at iteration based on “i”
    - v. Add “val” to back of “waveform”
    - vi. Increment “i”
  - d. End FOR
6. END WHILE
7. Call MakeWavFromVector function to convert vector to a sound file
8. Clear “waveform” from memory
9. RETURN 0

## PROGRAM GRAPH - CORE



## PROGRAM GRAPH - COMPLETION



## PROGRAM GRAPH – CHALLENGE

