Phase 1: finding creak values within segment boundaries

- skip either the adjusted file or the creak file until the filenames match

- create out file

- open both files

- get start and end time of current segment line

- adjusted files are formatted: [start time] [end time] [segment]

- creak files are formatted: [time samples],[binary value]

- if creak time is less than start time, (occurs when there is no segment during a certain time span, which does happen occasionally), skip to the next creak line

- if the creak time is within start time and end time for current segment, add the creak value to an array for that segment, then go to the next creak line

- if ther creak time is greater than end time, perform phase 2 for the current segment's creak array and then move on to the next adjusted line

Phase 2: creating 15 interval creak values within segment boundaries

- if there are 0 creak values in the segment range (segment would have to be less than 0.05 seconds long, but there are instances of this being the case), then take the previous creak value and set it as the overall average as well as each of the 15 interval averages for that segment

- if there is only 1 creak value in the segment range, then then take the single creak value and set it as the overall average as well as each of the 15 interval averages for that segment

- otherwise:

- get an interval value equal to 1/15 of the segment range

- set the creak value that is closest in time to the start of that interval as the value for that interval