TITLE

LAB 5

SECTION C

Kenneth Jacobson

SUBMISSION DATE:

4/10/2017

Problem

The problem in this lab was to create programs to average out values in a .dat file to try and clear up some noise from the original files. We were given white and pink noise to work with and told to create 3 programs with sampling values of 5,7, and 9 respectively.

Analysis

There isn't much to analyze with these programs as we didn't have any large problems to solve.

Design

We had a minimal hand in the design of these programs as a majority of the code was already written for us. The part we had to design was to define a constant variable for the number of samples and define a number of sample variables based on that constant along with an average variable to store our averaged values.

Testing

My initial code didn't initialize the sample variable and this lead to the first few avg values to be nan but after initializing all my sample variables the semantic error went away and I had finished the lab.

Comments

Some comments.

Source Code

Program with 5 samples

```
#include <stdio.h>

//declare the number of samples (N = 5) below
//use const or #define
#define N 5;

int main(void){
        //Your variable declarations below

        //1. Variables to store N = 5 samples
        //declare sample0, sample1, ..., sample4 below
        float sample0= 0.0, sample1= 0.0,sample2= 0.0,sample3= 0.0,sample4= 0.0;

        //2. declare variable to compute the average of N samples
        float avg;

        //Any other variable declarations needed by you
        //should be below



/******  DO NOT MODIFY BELOW LINES ********/

  //Current time read from input file
  float curTime = 0.0;
  //current sound sample read from input file
  float curSample = 0.0;

        //buffer size for fgets
 const int MAX_STR_SIZE = 100; // DO NOT CHANGE THIS LINE
 char metadata[MAX_STR_SIZE];  // DO NOT CHANGE THIS LINE

   // Scan in and print out metadata lines to the output file
   fgets(metadata, MAX_STR_SIZE, stdin);
   printf("%s",metadata);
   fgets(metadata, MAX_STR_SIZE, stdin);
   printf("%s",metadata);
/*****************************************/

        // While we have more lines remaining in the input sound sample file
        // feof - test for end of file, until no more samples to read
   while (!feof(stdin)) { // use this while loop to read each line of the .dat file

                //Your code to update sampleX variables
                //Move the contents of sample3 to sample 4 and so on
                //Move the current sample to variable sample0
                        sample4 = sample3;
```

```
                        sample3 = sample2;
                        sample2 = sample1;
                        sample1 = sample0;

                //Read the current time and the current sound sample during each
                //iteration through the while loop
                //In every iteration, you will get one sample and the associated time
    scanf("%f %f", &curTime, &curSample);
                sample0 = curSample;

                //Your logic to compute the average of the 5 samples and print
                //to output file
                        avg = (sample0+sample1+sample2+sample3+sample4)/N;

   // Print out the average to output file
    printf("%0.10f %0.10f\n", curTime, avg);

        }
}
```

Program with 7 samples

```
#include <stdio.h>

//declare the number of samples (N = 5) below
//use const or #define
#define N 7;

int main(void){
        //Your variable declarations below

        //1. Variables to store N = 5 samples
        //declare sample0, sample1, ..., sample4 below
        float sample0= 0.0, sample1= 0.0,sample2= 0.0,sample3= 0.0,sample4= 0.0;
        float sample5= 0.0,sample6= 0.0;

        //2. declare variable to compute the average of N samples
        float avg;

        //Any other variable declarations needed by you
        //should be below


/******  DO NOT MODIFY BELOW LINES ********/

  //Current time read from input file
  float curTime = 0.0;
  //current sound sample read from input file
  float curSample = 0.0;

        //buffer size for fgets
 const int MAX_STR_SIZE = 100; // DO NOT CHANGE THIS LINE
 char metadata[MAX_STR_SIZE];  // DO NOT CHANGE THIS LINE
```

```
   // Scan in and print out metadata lines to the output file
   fgets(metadata, MAX_STR_SIZE, stdin);
   printf("%s",metadata);
   fgets(metadata, MAX_STR_SIZE, stdin);
   printf("%s",metadata);
/***************************************/

        // While we have more lines remaining in the input sound sample file
        // feof - test for end of file, until no more samples to read
   while (!feof(stdin)) { // use this while loop to read each line of the .dat file

                //Your code to update sampleX variables
                //Move the contents of sample3 to sample 4 and so on
                //Move the current sample to variable sample0
                        sample6 = sample5;
                        sample5 = sample4;
                        sample4 = sample3;
                        sample3 = sample2;
                        sample2 = sample1;
                        sample1 = sample0;

                //Read the current time and the current sound sample during each
                //iteration through the while loop
                //In every iteration, you will get one sample and the associated time
    scanf("%f %f", &curTime, &curSample);
                sample0 = curSample;

                //Your logic to compute the average of the 5 samples and print
                //to output file
                        avg = (sample0+sample1+sample2+sample3+sample4+sample5+sample6)/N;

   // Print out the average to output file
    printf("%0.10f %0.10f\n", curTime, avg);

        }
}
```

Program with 9 samples

```
#include <stdio.h>

//declare the number of samples (N = 5) below
//use const or #define
#define N 9;

int main(void){
        //Your variable declarations below

        //1. Variables to store N = 5 samples
        //declare sample0, sample1, ..., sample4 below
        float sample0= 0.0, sample1= 0.0,sample2= 0.0,sample3= 0.0,sample4= 0.0;
```

```
        float sample5= 0.0,sample6= 0.0,sample7= 0.0,sample8= 0.0;

        //2. declare variable to compute the average of N samples
        float avg;

        //Any other variable declarations needed by you
        //should be below


/******  DO NOT MODIFY BELOW LINES ********/

  //Current time read from input file
  float curTime = 0.0;
  //current sound sample read from input file
  float curSample = 0.0;

        //buffer size for fgets
  const int MAX_STR_SIZE = 100; // DO NOT CHANGE THIS LINE
  char metadata[MAX_STR_SIZE];  // DO NOT CHANGE THIS LINE

   // Scan in and print out metadata lines to the output file
   fgets(metadata, MAX_STR_SIZE, stdin);
   printf("%s",metadata);
   fgets(metadata, MAX_STR_SIZE, stdin);
   printf("%s",metadata);
/*****************************************/

        // While we have more lines remaining in the input sound sample file
        // feof - test for end of file, until no more samples to read
   while (!feof(stdin)) { // use this while loop to read each line of the .dat file

                //Your code to update sampleX variables
                //Move the contents of sample3 to sample 4 and so on
                //Move the current sample to variable sample0
                        sample8 = sample7;
                        sample7 = sample6;
                        sample6 = sample5;
                        sample5 = sample4;
                        sample4 = sample3;
                        sample3 = sample2;
                        sample2 = sample1;
                        sample1 = sample0;

                //Read the current time and the current sound sample during each
                //iteration through the while loop
                //In every iteration, you will get one sample and the associated time
    scanf("%f %f", &curTime, &curSample);
                sample0 = curSample;

                //Your logic to compute the average of the 5 samples and print
                //to output file
                        avg =
(sample0+sample1+sample2+sample3+sample4+sample5+sample6+sample7+sample8)/N;
```

```
    // Print out the average to output file
      printf("%0.10f %0.10f\n", curTime, avg);

            }
}
```

Screen Shots

<Number the screenshots and paste here. The point of numbering the screenshots is so that you can refer to them during your discussion in the various parts above. Alternatively, you can include the screenshots in-line with the text above as part of your discussion.>