Filtering Noise With Arrays

LAB 7

SECTION C

Kenneth A. Jacobson

SUBMISSION DATE:

27.10.17

Problem

Our problem in this lab was to modify our code from Lab 5 to utilize arrays and to initialize the array with the first N values before beginning to print averaged values.

Analysis

This lab was rather simple as a majority of the code was already written and there was minimal need to write new code.

Design

The one part of the code that I did have to add was 2 for loops to move values in the array and to sum the array before averaging.

Testing

I didn’t have to do much testing as I have already completed this before and didn’t run into any bugs while coding.

Comments

NA

Source Code

5 Samples:

```
//Author: Kenneth A. Jacobson
//Description: Average sound values in a .dat file with N sample
//Lab 7
//20.10.2017

#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 sample[5]= {0.0,0.0,0.0,0.0,0.0};


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


        //Any other variable declarations needed by you
        //should be below
        float sum;      int i;    int j;
        int ceil= N-1;
        int k= 1;


/****** 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
        for(i=ceil; i>0; i--){
            sample[i]= sample[i-1];
        }

    //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);
        sample[0] = curSample;

    //Calculate average and print it to file with current time
    //after first N values have been saved into the sample[] array
        if(k>=ceil){
            for(j=0; j<=ceil; j++){
                sum += sample[j];
            }
            avg = sum/N;
            sum = 0.0;
            printf("%0.10f %0.10f\n", curTime, avg);
        }
        k++;
    }
}
```

7 Samples:

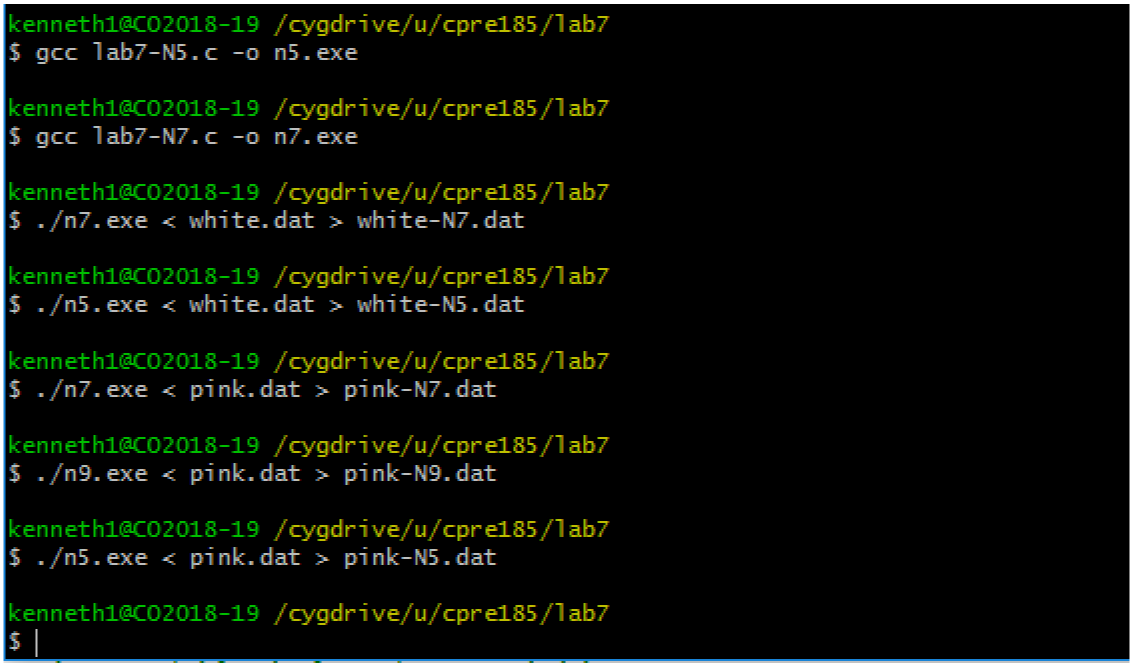
```
//Author: Kenneth A. Jacobson
//Description: Average sound values in a .dat file with N sample
//Lab 7
//20.10.2017

#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 sample[7]= {0.0,0.0,0.0,0.0,0.0,0.0,0.0};

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

        //Any other variable declarations needed by you
        //should be below
        float sum;      int i;    int j;
        int ceil = N-1;
        int k= 1;

/****** 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
                for(i= ceil; i>0; i--){
                        sample[i] = sample[i-1];
                }

        //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);
                        sample[0] = curSample;

        //Calculate average and print it to file with current time
        //after first N values have been saved into the sample[] array
                if(k>=ceil){
                        for(j=0; j<=ceil; j++){
                                sum += sample[j];
                        }
                        avg = sum/N;
                        sum = 0.0;
                        printf("%0.10f %0.10f\n", curTime, avg);
                }
                k++;
        }
}
```

9 Samples:

```
//Author: Kenneth A. Jacobson
//Description: Average sound values in a .dat file with N sample
//Lab 7
//20.10.2017

#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 sample[9]= {0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0};

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

        //Any other variable declarations needed by you
        //should be below
        float sum;                              int i;
        int ceil= N-1;                  int j;
        int k= 1;

/******  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 sample[X] variables
        //Move the contents of sample3 to sample 4 and so on
        //Move the current sample to variable sample0
                for(i=ceil; i>0; i--){
                    sample[i] = sample[i-1];
                }
        //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);
                    sample[0] = curSample;

                //Calculate average and print it to file with current time
                //after first N values have been saved into the sample[] array
                if(k>=ceil){
                    for(j=0; j<=ceil; j++){
                        sum += sample[j];
                    }
                    avg = sum/N;
                    sum = 0.0;
                    printf("%0.10f %0.10f\n", curTime, avg);
                }
                k++;
        }
}
```

Screen Shots

Program Compiling:



```
kenneth1@CO2018-19 /cygdrive/u/cpre185/lab7
$ gcc lab7-N5.c -o n5.exe

kenneth1@CO2018-19 /cygdrive/u/cpre185/lab7
$ gcc lab7-N7.c -o n7.exe

kenneth1@CO2018-19 /cygdrive/u/cpre185/lab7
$ ./n7.exe < white.dat > white-N7.dat

kenneth1@CO2018-19 /cygdrive/u/cpre185/lab7
$ ./n5.exe < white.dat > white-N5.dat

kenneth1@CO2018-19 /cygdrive/u/cpre185/lab7
$ ./n7.exe < pink.dat > pink-N7.dat

kenneth1@CO2018-19 /cygdrive/u/cpre185/lab7
$ ./n9.exe < pink.dat > pink-N9.dat

kenneth1@CO2018-19 /cygdrive/u/cpre185/lab7
$ ./n5.exe < pink.dat > pink-N5.dat

kenneth1@CO2018-19 /cygdrive/u/cpre185/lab7
$ |
```