

Name: _____

Problem	Points	Score
1	50	
2	50	
Total	100	

Notes:

- (1) For this exam you are allowed to open a terminal window on your computer, you are allowed to web surf with Google, but you cannot use online chat or other interactive services.

The first step in this exam is to create a workspace in the following directory:

`/data/courses/ece_1111/current/exams/exam_02`

Your directory should be your last name all lowercase (e.g, “picone”). Set the permissions using “`chmod u+rwx,g-rwx,o-rwx <lastname>`” so only you have read and write permission to this directory. Create two subdirectories within this directory: `p01` and `p02`. You will use these for problems 1 and 2 respectively. Put ALL your code in these directories.

Problem No. 1: In HW #7, we learned how to read a file using frames and windows. Modify your program so that it reads a binary file of short integers using a frame size of M samples and a window size of W samples. Your interface must be the following:

`p1.exe <filename> M W`

Use this file for testing:

`/data/courses/ece_1111/current/exams/exam_02/picone/p1.dat`

This file contains short integers ranging in value from -11 to 10:

```
necd_999_[1]: od -s p1.dat
0000000  -11  -10  -9   -8   -7   -6   -5   -4
0000020   -3   -2   -1    0    1    2    3    4
0000040    5    6    7    8    9   10
0000054
```

There are two things your program must do differently:

- (1) The window, instead of being center-aligned, must be left-aligned. This means the first window should be the first W samples in the file. The second window must be this window shifted by M samples. The third window must be the second window shifted by M samples. For example, for `p1.dat` for $M=2$ and $W=4$:

```
frame 1: [-11 -10 -9 -8]
frame 2: [-9 -8 -7 -6]
frame 3: [-7 -6 -5 -4]
```

Your code must work for all combinations of M and W and different file lengths, and must handle the end of file condition properly (values beyond the end of file are assumed to be zero). You must handle the case where $W < M$.

You must only read M samples with each iteration after initialization. You cannot simply position the file pointer and read W samples each time. The total number of samples you read must be equal to M so that you are not doing unnecessary I/O.

- (2) You must multiply each window of data, s_w , and a weighting function, h , where h is a vector of length W whose values are all 2. Then you must sum the squares of the result to compute “energy”. Print the following information to stdout: input filename, M , W , and for each frame, the frame index and the energy value.

Note that the code I provided as an example will not work for this case. You must modify that code. Simply turning in that code will get you a score of zero on this problem.

Problem No. 2: Create a program that does the following:

- (1) Calls a function named `create_matrix(m, N, M)`. This function allocates memory to hold a matrix of dimension $M \times N$. Memory for the matrix must be allocated within this function. You can use floats, doubles, longs or whatever numeric data type you prefer. Just be consistent.

- (2) Calls a function named `set_matrix(m, value)` that sets all the values in the matrix `m` to `value`.
- (3) Calls a function named `print_matrix(FILE*)` that prints the contents of the matrix to `stdout`. `Stdout` should be an argument to the function in your main program.
- (4) Calls a function `delete_matrix(m)` that frees the memory space so there are no memory leaks.

Summary: Work directly in your workspace so that you don't lose any time transferring files. Your source code should be clean and commented. If I can't read it and understand it, your grade will suffer. Comment as you go along – it will help you understand what you are doing.