

CISC106 Fall 2008 Lab10

- Review the code examples from class.
- Some programs below are associated with a question. **Answer the questions** using comments below your code in the m-file.
- The office hours of the TAs and the instructor are on the class website. Visit us!
- **NOTE:** Every function comment section should contain, at a minimum, *three examples* of the function being called and the result of evaluating the call. These examples must include boundary conditions (as discussed in class). Your test files must cover *at least* these exact examples (otherwise, why did you choose them?) and possibly more. Testing is important.
- **Every** M-file or .c you write or modify must be demonstrated, either by running a script test file in a diary or by testing at the command line. Note that if you write function **foo.m** and test script **fooTest.m**, you can demonstrate both by running only fooTest.m (assuming fooTest works).

Problems

1. Write a Matlab program which calls various functions for each of the following tasks: a) Read an Image, given the file name, b) Mean filter the Image, given the window neighborhood, c) Median filter the Image, given the window neighborhood (for this, use the BubbleSort function discussed in the class), d) Flip left-to-right, e) Flip top-to-bottom. For each of these tasks, show both the original and processed image in a panel. NOTE: All these programs were discussed in class, and are available in Sakai notes. But you need to write them as functions and your main program calls these functions.

Make sure to use 'imfinfo' to get the image dimensions.

2. Copy the hello.c file from the class website into your lab10 directory. At a Unix command prompt (not in Matlab) do the following: determine which compiler(s) you have available in your path¹. Type

```
> which CC
```

(case is important) and then type

```
> which g++
```

The Unix which command will either return a path (characters separated by slashes) or "command not found". If neither is found, consult your TA for how to set up your .localenv file so that you can use a compiler, or read about it at

<http://www.udel.edu/topics/os/unix/startup-files/csh/index.html>

You can also use a compiler by typing its whole path name, but that gets old quickly. So now I'll assume you have the CC compiler available. In your X terminal type:

¹The first, (CC) is a commercial compiler, supplied by Sun Microsystems. The second, g++ is an open source compiler, supplied by the GNU project, which is staffed by volunteer programmers.

```
> CC hello.cc
```

Now use the 'ls' program to see your directory contents. You will see both hello.c and a new file, a.out² which is an executable file. Run it by typing

```
> ./a.out
```

You are now ready to use the 'script' command to make a record of your work. Note that you must be careful when using the script command. The script command will wipe out your work if you are not careful. Script works by writing into whatever filename you specify, so don't specify a filename that has good stuff in it (like your program name). The thing to remember is: on the command line, type 'script' followed by the name of a .txt file; for example, type these commands:

```
> script lab10.1.txt
> ls
> exit
```

Now look at the file lab10.1.txt using pico; you will see the results of your 'ls' inside it. The diary command in Matlab was derived from the same idea. Never put script followed by the name of a .c file. It will wipe out the .c file (where your code is!). Type 'script lab10.1.txt'. Script is going to write over the file you already created. Then go through the following steps in the same xterm:

- list your program (type 'cat hello.c' to do this.)
- compile your program (type 'CC hello.c' or 'g++ hello.c' to do this)
- run your program (type './a.out')
- exit from the script (type 'exit').

You will script all of your C programs, so make sure you understand these steps and can repeat them.

If your TA requires a paper copy, be sure that you have a printed copy of your function M-files, script M-files, image files, diary files, C files, script files demonstrating your testing. All must be stapled together, with your name and lab section on the top page.

Be sure that you upload a copy of all the MATLAB functions, script, images, and diary files to Sakai. Then, click submit ONLY ONCE to send these to your Sakai and your TA.

On the first page of every printed copy for this course, your name, section, and TA's name must appear.

²Think: What does ./mean?