Lab 4 Report Instructions

Makefile

- Compilation and linking
- Testing

References:
Compiling a C program:

```
$ gcc prog.c {o prog
```

`https://github.com/yunghsianglu/IntermediateCProgramming`
Make sure you create a separate directory for Exercise 2 and store all the files and subdirectories there.

Exercise 1

1.1

Prepare two data sets (files), one with unique integer numbers and the other, similar to the first one, but with some numbers repeated. Then, without using a Makefile, compile and run the `areDistinct` application from Chapter 5, pp. 66-68.

Submit:

- The printout showing the data set with no repeated numbers
- The screenshot of the output from your program on the first data set
- The printout showing the data set with some numbers repeated
- The screenshot of the output from your program

Exercise 2

2.1

Prepare a Makefile with three tests (use the same data sets as in Exercise 1) and similar to the file shown on page 74. Create three directories: `inputs`, `outputs` and `expected`, as shown in the example on page 74. The first two tests should have the results from Exercise 1 stored in the `expected` directory and should complete successfully. The third test is to be designed to fail (there are two simple ways to have the third test run fail: you may put incorrect values in the expected directory, or you may use a different version of the program for the third test that does not work correctly).

First, run:

```
$ make clean
```

to delete the executable and all the files in the outputs directory.

Then, follow these steps:

- Show the directory structure and all the files of your project, for example:

  ```
  ls -R -l
  ```

  (Note: the `-R` option prints the contents of all subdirectories.)

- Show the contents of the `expected1`, `expected2`, `expected3` and `output1`, `output2`, `output3` files for all three tests
- Show the contents of the makefile
- Run `make` to compile your program
- Show all the directories and all the files again
- Show the timestamps for one of the source code `.c` files, the created corresponding object code `.o` file and for the executable file
- Use `make` to run all the tests
- Show the output from `make`, if any
- Show all the directories and all the files again

Exercise 3

3.1

In a new directory, copy the program files from your Assignment 2. Modify the program from A2 so that it outputs the HTML tags with the "<" and ">" characters around them (i.e. instead of outputting "HTML" your program should print "`<HTML>`"). Create a makefile similar to the file from the example on page 74, but with different tests. Your new makefile will have to compile and then run the `htag1` program twice, reading and writing data from/to a file using the input/output redirection.

The first run (`test1`) should read the input from a sample HTML file (any sample file with HTML tags will do), and the second run (`test2`) should read from the file generated as the output from processing of the first file. Then the makefile should execute `diff`, comparing the outputs from the two runs (i.e. instead of having two files for comparison `output1` and `expected1` as in the textbook, you will compare `output_from_first_run` and `output_from_second_run`). You should verify with `diff` if the output from the first run is the same as the output from the second run, i.e. when the second run processes the output from the first run.

Submit:

- Show the directory structure and all the files of your project
- Show the contents of the makefile
- Run the tests (the `make` should compile the files automatically)
- Show the output from `make`, if any
- Inspect all the files and comment on the result from your tests

Submission Instructions

Type (collect the output, etc.) the answers to Questions 1.1, 2.1, and 3.1 (use the headings indicating the question number) into a word processor and store them in a single PDF file. Include a preamble (title, section) with your name, student number, course number, etc. Please number all the pages!

Submit this **SINGLE PDF file** on D2L as the **Lab 4 Report** by the due date indicated in the D2L Dropbox.