CSCI 352 Unix Software Development
Summer 2015
C Programming Warm-up Task

The purpose of this warm-up task is to get you started in C programming on a fairly simple task, but one that builds a component that will be needed for the first assignment. You may, but are not required to, submit your program for evaluation and critique. You are encouraged to submit your program multiple times so that you may improve it from feedback.

fref "find regular expression in files"

fref is a utility that will search through one or more files and report any lines in those files that contain a string matching a specified regular expression.

Regular Expressions

For this exercise (and Assignment 1) we will use a simple regular expression language:

Character	Meaning
x	Matches single specified character x
. (period)	Matches any single character
^	Matches the beginning of the string
$	Matches the end of the string
*	Matches zero or more occurrences of the previous character

The regular expression "^ *}" will match lines that start with a } following zero or more leading spaces.

You are provided with the source code and header file for functions which perform the regular expression match. These functions are a very nice example of the use of C – you should study them carefully and ensure that you understand them. The functions are described in Chapter 1 of the book "Beautiful Code", by Oram & Wilson. That entire book presents elegant programming solutions to a variety of problems using a variety of programming languages.

As described in the header file, function match() is the function used to determine whether a regular expression matches a substring of a provided string.

```
int match ( char *regexp, char *text);
```

Returns 1 if regexp is found anywhere in text, 0 otherwise.

Specification of fref

- The fref program must get a regular expression and a list of file names from the command line.
- If there are fewer than 3 command line arguments, fref is to display a usage message and terminate.
- For each line from one of the specified files that contains a string matching the regular expression, fref is to display the file name, line number and the line. For example:

```
> ./fref "^ *}" regexp.c
regexp.c:35     } while (*text++ != '\0');
regexp.c:37 }
regexp.c:60 }
regexp.c:77     } while (t-- > text);
regexp.c:80 }
```

  In this example, we look for the regular expression `^ *}` in the file regexp.c. That will match any line that starts with zero or more spaces, followed by }. Notice that the regular expression is enclosed in quotes – this is to prevent bash from interpreting "*" as a wildcard character and using the space in the regular expression as a command line argument separator.

- For any file specified on the command line which cannot be opened for input, fref is to display a message "File xxx not found".

Coding Standards

1. Use meaningful names that give the reader a clue as to the purpose of the thing being named.
2. Avoid the repeated use of numeric constants. For any numeric constants used in your program, use #define preprocessor directives to define a macro name and then use that name wherever the value is needed.
3. Use comments at the start of the program to identify the purpose of the program, the author and the date written.
4. Use comments at the start of each function to describe the purpose of the function, the purpose of each parameter to the function, and the return value from the function.
5. Use comments at the start of each section of the program to explain what that part of the program does.
6. Use consistent indentation.