

11.5 Practice Programs

Sample solutions to these problems will be included by the beginning of 2013 at the following website:

<http://www.cs.ucf.edu/~dmarino/ucf/transparency/cop3223/book/>

1) Write a function that takes in a string and a character, and returns the number of times that character appears in the string. The prototype for the function is given below:

```
int num_appearances(char word[], char letter);
```

2) Write a function that takes in a string and changes each of its alphabetic characters to lowercase. Use the macro `toupper` in `ctype.h`. The prototype for the function is given below:

```
void to_lower_case(char word[]);
```

3) Write a function that takes in a string and returns 1 if the string is a palindrome and 0 otherwise. A palindrome is a string that reads the exact same forwards and backwards. ("racecar" and "madam" are palindromes.) Make comparisons between letters to be case insensitive. The prototype for the function is given below:

```
int is_pal(char word[]);
```

4) Write a function that takes in two strings, and determines whether or not the second string appears as a substring in the first. A substring is a sequence of consecutive characters within a string. For example, "newspaper" contains the substring "spa", but doesn't contain the substring "nap". The prototype for the function is given below:

```
int has_substring(char line[], char str[]);
```

5) Edit the function for question #4 so that it returns the index of the beginning of the first substring match, if it exists, and returns -1 otherwise. Thus, `substring_at("hello", "lo")` should return 3 while `substring_at("newspaper", "nap")` should return -1. The prototype for the function is given below:

```
int substring_at(char line[], char str[]);
```

6) Write a function that takes in two strings and returns a match score to indicate how similar the words are. If the two strings are different lengths, then your function should return 0. Otherwise, if the two strings are the same length, the score returned should be the number of characters in corresponding locations that are the same. For example, the match score between "home" nad "host" should be 2 and the match score between "paper" and "caper" should be 4. The prototype for the function is given below:

```
int match_score(char word1[], char word2[]);
```

7) Write a program that reads in 5 uppercase alphabetic strings from the user and determines whether or not the 5 strings were entered in alphabetic order.

8) Write a program that reads in a dictionary of words from the file "dictionary.txt" and then asks the user to enter a string. Determine whether that string is in the dictionary or not and print a corresponding message. The file format of "dictionary.txt" is that it contains a single positive integer, n , on its first line, indicating the number of words in the dictionary. The following n lines have one word each. Each word will only contain lowercase alphabetic characters.

9) Using your program from #8, after your program reads in the dictionary, ask the user to enter in the name of a text file that only contains lowercase words. You may assume that this file only contains valid words stored in the dictionary. Count how many occurrences of each word are in the second text file. Print out a listing of each word and how many times it occurs, to the screen. For example, if the text file stored: "this book is a great book to read", then your program should print out:

a	1
book	2
great	1
is	1
read	1
this	1
to	1

10) Write a program that reads in two lists of words sorted in alphabetic order from a file and prints out to the screen all the words that appear in both lists in alphabetic order. The input file will start with a line with a single positive integer, n , on a line by itself, indicating the number of words in the first list. The next n lines will contain the words, all lowercase. This is followed by a line contain a single positive integer, m , indicating the number of words in the second list. The next m lines will contain the words in this second list, also all lowercase.