COMP 30400 – Programming I & II
Lab 5 – 4 hours

Practical Overview

This practical is for Lecture 5. All portfolio programs require a header at the start of every source file. Refer to lecture notes on how to format it. Comments are expected throughout the code, where relevant. They should be used to discuss motivations for your approach and alternative ways of implementing the solution where appropriate.

Appropriate error checking for memory allocations, and free-ing of memory is required where relevant.

When using malloc(), you must #include <stdlib.h>

There is an example source code file that accompanies this lab sheet.

1) Portfolio Program 23
Based on program 19 (the guess the word program), modify the program so that it uses a function to wait and read the user's input.

Hints:
1) Have a look at slide 12 of the lecture notes – it gives an example of declaring and using a function.
2) Functions only see variables that are defined within that function. For example, variables you use in your main() function will not exist in your other functions. This is why you must pass them by using parameters.
3) void get_user_input(char* array);
4) You can treat char* and char[] as the same type.

2) Portfolio Program 24
Based on program 21 (the reverse the words program), add a function to the program that reverses the characters for an individual word. The program should use the following function declaration:

void swap_characters_in_word(char *words, int start_of_word);

The start_of_word variable should contain the index of the first character in the word in which it is to reverse.

The program should provide the same functionality as program 21, but by using the function described above.

Hints: There are several approaches to this problem. You could allocate a character array/block of memory and store the swapped characters there and then copy the word back using strcpy(). Another solution is to use a temporary character while performing the swap. Where:

```
char temp = array[ index_of_last_letter_in_word ];
array[index_of_last_letter_in_word] = array[index_of_first_letter_in_word];
array[index_of_first_letter_in_word] = temp;
```

Both approaches have different advantages/disadvantages.

Hints:
swap_characters_in_word(input_array, element_index_of_first_letter_in_word);

3) Portfolio Program 25
Write a program that lets the user input any number of words. Count how often each alphabetic character occurs in the input, and print out the results, such as:

a = 3
b = 0
c = 2
d = 4
…
…
…

Hints:
Use an int[] which is large enough to store all letters in the alphabet.
Use a for loop to set everything to zero.
Use a for loop to count the occurrence of each character in the input array.

4) Portfolio Program 26

Based on the example program in Lecture 5 slide number 28, copy the program into your own source file and compile it. Try running it with different options and look at the outputs. Remember to #include <stdio.h>.

5) Remember to backup your portfolio.