COMP 30400 – Programming I & II
Lab 6 – 2 hours

Practical Overview

This practical is for Lectures 5 and 6. 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.

1) Portfolio Program 27

Write a program that allocates a small amount of memory using malloc. Using fgets, let the user input some words. Then, use printf to output the character array, and the memory address of it.

In your comments, you should explain why printf does not print the contents of character array twice. We will review this in the next lecture.

Hints:

```
char *memory;
memory = (char*) malloc(50); /*use a const instead of 50 */
fgets(memory, 50, stdin);
printf("The character array starting from memory location %u, contains: %s\n", memory, memory);
```

2) Portfolio Program 28

Based on portfolio program 27, use a character array (such as char input[50];) instead of malloc(). When you run the program, printf() will perform the same as program 27 – the memory address may change, but the array has an address too, just as the pointer did. You should highlight this in your comments.

3) Portfolio Program 29

Based on program 13, and program 26, write a calculator program where the user specifies the options on the command line. For example:

Calculator.exe 5 + 1234
Calculator.exe 5 x 2

NOTE: Use the x character for multiply instead of *.

This program should not loop like program 13 did.

Hints: You'll need to use the second main() for this.

To convert the character array to an integer, you can use **sscanf**() – which works in the same way of scanf, but it lets us process on a string:

sscanf(character_array, "%d", &number);

This will make scanf process the variable "character_array" instead of what the user enters into the terminal window.

4) Portfolio Program 30

1. Write a program that uses malloc to allocate a character array of 50 characters named **word**.
2. Use fgets to input a word from the user (store the input in **word**).
3. Create a pointer to a character array, named **ptr**.
4. Make **ptr** point to the second character in the array named **word**.
5. Modify the 2nd element in **ptr** to be a '_' (underscore) and the 4th element to be a '$' (dollar). Use printf to display the contents of **word**.

Note: The final output should not (exactly) match the user's input.

5) Remember to backup your portfolio.