

## Programming – DT282/1 & DT228/1

### Lab 11 – Tuesday, January 24<sup>th</sup>, 2017

**Note:** You are expected to finish all programmes in your own time if you do not get these done during the lab session. This is your own responsibility.

### Dynamic Memory Allocation (DMA)

**Remember:** Use Symbolic names in your programs. Do not hard-code.

Write separate programs to:

1. Write the 2 programs covered in lecture class for both `malloc()` and `calloc()`. After you run these programs, change the size of the allocated memory to be smaller than required (i.e. remove the `sizeof(int)` and replace it with a hardcoded integer number, as we discussed in lecture class). Compile and run the changed programs. What happens?
2. Chapter 9 – Q9, Q10, Q11 (see below)
3. Write a program that uses dynamic memory allocation to allocate memory for 5 floating-point numbers. You can use either `malloc()` or `calloc()`.

After memory has been allocated for the 5 float numbers, enter these numbers. Calculate the average of these numbers and store this average in another allocated memory block. Display all of the 5 float values on the screen and the average. (Hint: you will need to use 2 float pointers, one pointing to the block of memory storing the 5 floating-point numbers, and one pointing to the block of memory storing the average of the 5 numbers).

**Note:** Be very careful with your use of pointers in these questions. Try not to exceed the bounds of the arrays when moving the pointers.

9. Using `malloc()` or `calloc()`, write a program to input a specified number of integer values into an array and to display the array and the sum of the elements in the array. Use pointers, not subscripts, in the program.

10. Given an array such as

```
char chars[] = { 'a', ' ', 'b', ' ', 'c', ' ', ' ', 'd' } ;
```

write a program that replaces all the blank elements in a character array with the underline character '\_'. Use a pointer, rather than a subscript, to access the elements of the array.

11. Given the following arrays,

```
float litres[] = { 11.5, 11.21, 12.7, 12.6, 12.4 } ;  
float miles[] = { 471.5, 358.72, 495.3, 453.6, 421.6 } ;  
int mpl[5] ; /* Miles per litre. */
```

write a program to calculate and display the value of each element of `mpl`. Use pointers, rather than subscripts, to access the elements of each array.