



# COMP1511

## Tutorial 4

*Arrays | Functions | scanf loops*

# Arrays and Loops





# Arrays Round Robin

We'll be completing 2 procedures on the whiteboards in your groups:  
*odd\_only* and *copy\_array*

You may only send up 1 person at a time

Tasks must be split evenly - you can't send someone up to do a second task until everyone's done at least one task

## odd\_only

- Create an integer array with at least 5 elements.
- Create a while loop which loops through every element of the array. You will also need a variable above the while loop.
- Write an if statement which adds one to variables in the while loop if they are even.
- Write another while loop which goes through the array with a different iterator
- Print out the values in the updated array.

## copy\_array

- Create an array of doubles with at least 3 elements.
- Create another double array of 100 elements; but every element initialised to 0.
- Create a while loop which loops through every element of the first array.
- Copy the elements of the first array into the second array (leave 0's at the end)
- Create a while loop which prints out all the elements of the second array.



# More Functions

We'll be writing functions that use this struct and enum to deal with colours:

*make\_colour*

*get\_main\_colour*

*brightness*

*average\_brightness*

```
struct colour {  
    int red;  
    int green;  
    int blue;  
};
```

```
enum main_colours {  
    RED,  
    GREEN,  
    BLUE,  
};
```



# scanf loops

We're going to make a program that calls *scanf* multiple times to scan in multiple values from the user. This program will:

- Scan velocity values into an array

- Calculate the average of the last 5 values

What are the main steps we'll need to make this program?



# scanf loops

What does *scanf* return?

How can we use this result to check whether a user signals the end-of-file?

This lets us use *scanf* in a loop to scan in multiple values!

*scanf* loop demo

