



Lecture 7

An array of arrays – how to go 2D



IN WEEK 3, WE...

- Talked about the importance of style – work neatly as you go!
- Discovered functions (separate chunks of code for reuse, help to segment the problem)
- Got introduced to arrays – homogenous collections – stores the same type of variable in a collection



TODAY...

- Assignment 1 is all systems go!
- A quick tangent to talk about C libraries
- Array of arrays - 2D arrays

WHERE IS THE CODE?

**LIVE LECTURE CODE CAN BE
FOUND HERE:**

<https://cgi.cse.unsw.edu.au/~cs1511/21T3/live/Week04/>

ASSIGNMENT 1

THE WAIT IS OVER



- Assignment 1 has been released
- CSE Valley - we have returned back to farming ...
- Aims of the assignment
 - Apply arrays and two-dimensional arrays in solving problems
 - Apply good style to your code
 - Apply the use of functions in code
 - Practice skills in debugging code, and skills in patience as you search for one missing semi-colon

ASSIGNMENT 1

THE WAIT IS OVER



- The Assignment has 4 stages, each stage ramps up with difficulty (just like the lab exercises)
- Suggest going through the stages chronologically – do not skip stages
- Live Stream to go through the assignment in more detail:
 - Thursday 6:30pm
 - Link for the livestream:
<https://www.youtube.com/watch?v=QCgr8DHKDz8>

C LIBRARIES

GOOD FOR BORROWING A LOT OF FUNCTIONS



Good reference if you are interested in learning more about each library:

https://www.tutorialspoint.com/c_standard_library/index.htm

- C has a number of standard libraries available to us
- Libraries are usually .h files (header files)
- We can use these libraries whenever we want to borrow some functions by:
`#include <library_name.h>`
- So far we have used
 - `<stdio.h>` Standard Input/Output Library
- Other useful libraries we may have seen:
 - `<stdlib.h>` Standard Library
 - `<math.h>` Mathematics Library
- Sometimes we can just borrow functions instead of writing them from scratch, like `printf`, `scanf` etc.

A QUICK RECAP OF 1D ARRAYS

FROM LAST WEEK

Remember that arrays:

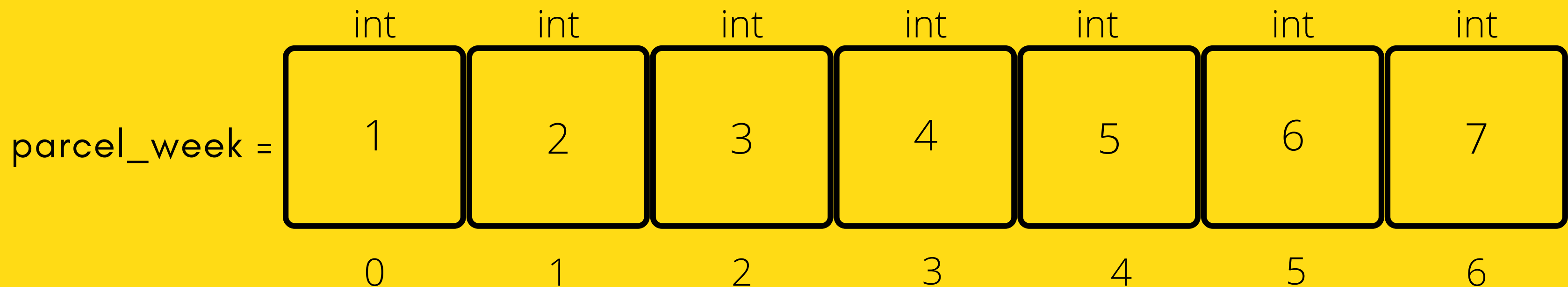
- a collection all of the same type
- declared by using a type, name and a size of the array
- you can easily access individual elements of an array by using an index
- Indexing starts at 0 and moves through until (size - 1) of the array
- go hand in hand with while loops that make it easy to work through an array

A QUICK RECAP OF 1D ARRAYS

FROM LAST WEEK

We looked at parcel deliveries in a specific week last time, for example, below is an array called `parcel_week` with seven elements initialised to contain values 1, 2, 3, 4, 5, 6, 7

```
int parcel_week[7] = {1, 2, 3, 4, 5, 6, 7}
```



A QUICK RECAP OF 1D ARRAYS

LET'S DO AN ARRAY PROBLEM TO REINFORCE 1D ARRAYS



Problem: A user is asked to enter 10 numbers. We will then go through these numbers and find the highest number and output what the highest number is to the user.

highest_number.c

YOU CAN HAVE AN ARRAY OF ANYTHING

EVEN AN ARRAY OF ARRAYS

What does an array inside an array look like?

- Think of it as a grid
- This means to access an element now you will need to `array[row][column]`
- Think of our parcel problem from last week where we kept track for one week, what if we now wanted to keep track for four weeks?

```
parcels[4][7] = {};
```

	col 0	col 1	col 2	col 3	col 4	col 5	col 6
row 0	1	2	3	4	5	6	7
row 1	8	82	13	41	15	16	17
row 2	21	22	23	42	25	26	27
row 3	31	32	33	34	35	36	37

**HOW DO YOU
THINK WE COULD
GO THROUGH THE
2D ARRAY TO
PRINT?**

**THINK BACK TO WEEK 2
PROBLEM WHERE WE
PRINTED A GRID...**

while inside a while
to print a grid

[Week02/print_grid.c]

```
int row = 1;
while (row <= size) {

    int col = 1;
    while (col <= size) {
        printf("%d ", col);
        col = col + 1;
    }
    printf("\n");
    row = row + 1;
}
```

HOW DO YOU THINK WE COULD GO THROUGH THE 2D ARRAY TO PRINT?

WE CAN TRANSLATE THIS TO OUR 2D ARRAY ALSO, SINCE IT IS A GRID

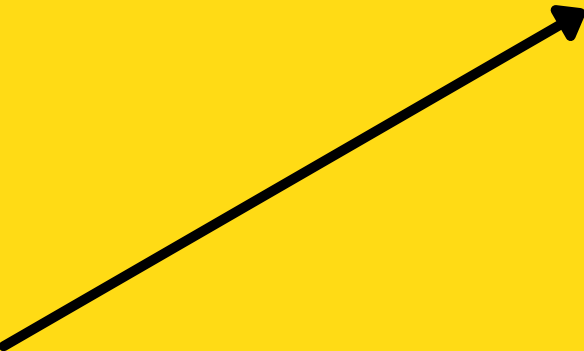
```
int row = 0;
while (row < size) {

    int col = 0;
    while (col < size) {
        printf("%d ", col);
        col = col + 1;
    }
    printf("\n");
    row = row + 1;
}
```

while inside a while
to print a 2D array

[Week04/print_array_grid.c]

```
int row = 0;
while (row < N_ROWS) {
    int col = 0;
    while (col < N_COLS) {
        printf("%d ", array[row][col]);
        col++;
    }
    printf("\n");
    row++;
}
```



LET'S DO A QUICK CODE DEMO OF IT

CODE, CODE, CODE!

print_2Darray.c

```
011110110110111110111000001111000001000001111100101010000000110000011101010101
11101011101001110100001111110110111101101100111101101110000111101010101000001111
01110111100110111000101111100000101110110101011110000001001011010010100111110101
1111011111111101100100101011001010101001111010101001000000010101000111111111010
11001101010011101010111010011010010100101110011001100001011110101000101010100101
01110110010110110000001101011110100001000100110111111111101110000111101000011110
11111101001100011001110001000110111000010001111011000000001101101001000111000001
1100110110101111110100100100100101110000110011011100110011000111001100010100110
00101010101000000100110101000100010101101011000000101001010001100011001010010001
001001101011011001010101100110101000110111000000001101001001000100000011110011011
01010101100011011011101000101001111110001000000100010010110000101110001110001100
11110010101001011011110011000011110000000110010100111110101100101100001000001110
10001101010100000000001111110010001000000111101000111101000010100100111100011001
11100101111101110011001100001010110001010110001110101110000111001100110101000011
01000000101101101111100000000011111100011101001001111101110110011110000010101001
10010100101011111100011100101101001100000111011001000111000010111011111010110101
1010000001111010110110011000001111100001010000110000111100011000000000010000101
110100100011011001000111001101110100100000101111110010010011111011110111101100000
01110111000011101000000001001000001000001000100111111110111111011101011001000010
1100011000101110100001100111101100010110101000000101011011010011101001000100100
11011001010000101111001000111001011110000010101101101001000110000010101010110010
010000100110101110010111111001101101100110110100111111111000100100101000000011000
001101001001101110111110100000011010111001011101010100011010100010011001001101
111001110001010000000101001111101100101010001001100001101011110101110100000110111
11011111111010000100010001101011100001110001110101100110101010111001101110100111
01111001101111011100100001100111111010110011000010110111100000011100000010000000
11110111010011101000010111010111011011101000010000101001110011110111100100010011
11100011010000110001011010111101011001110111100010011010000100001100100010110110
01110101011110111011111111010
```

BREAK TIME (5 MINUTES)

There are five bags of gold that all look identical, and each has ten gold pieces in it. One of the five bags has fake gold in it. The real gold, fake gold, and all five bags are identical in every way, except the pieces of fake gold each weigh 1.1 grams, and the real gold pieces each weigh 1 gram. You have a perfectly accurate digital gram scale and can use it only once. How do you determine which bag has the fake gold?

PROBLEM TIME

AN ARRAY INSIDE AN ARRAY IN ACTION



JJ_park_adventure.c

Problem: Jax and Juno have just found out that the 5km radius restraint has been removed. They are excited at the prospect of finally visiting different parks and exploring every corner of those parks. The smells, the adventures, the dirty water puddles – oh my! Each park is represented as a 10x10 grid. We will start with just Jax exploring the park, and then maybe Juno will join him (so for now just one dog!) Jax is able to move around each park (left, right, up and down) and wants to visit as many sections of the park as possible. Once he has explored the park, or barks that he wishes to go home or if he comes to a spot that he has already been to, it will be time to go home.

PROBLEM TIME

AN ARRAY INSIDE AN ARRAY IN ACTION



- Like Assignment 1, we have starter code that we will begin working with
- The starter code provides us partial functionality for printing out our square park grid
- Let's move over to gedit and start solving this problem by breaking it up into components

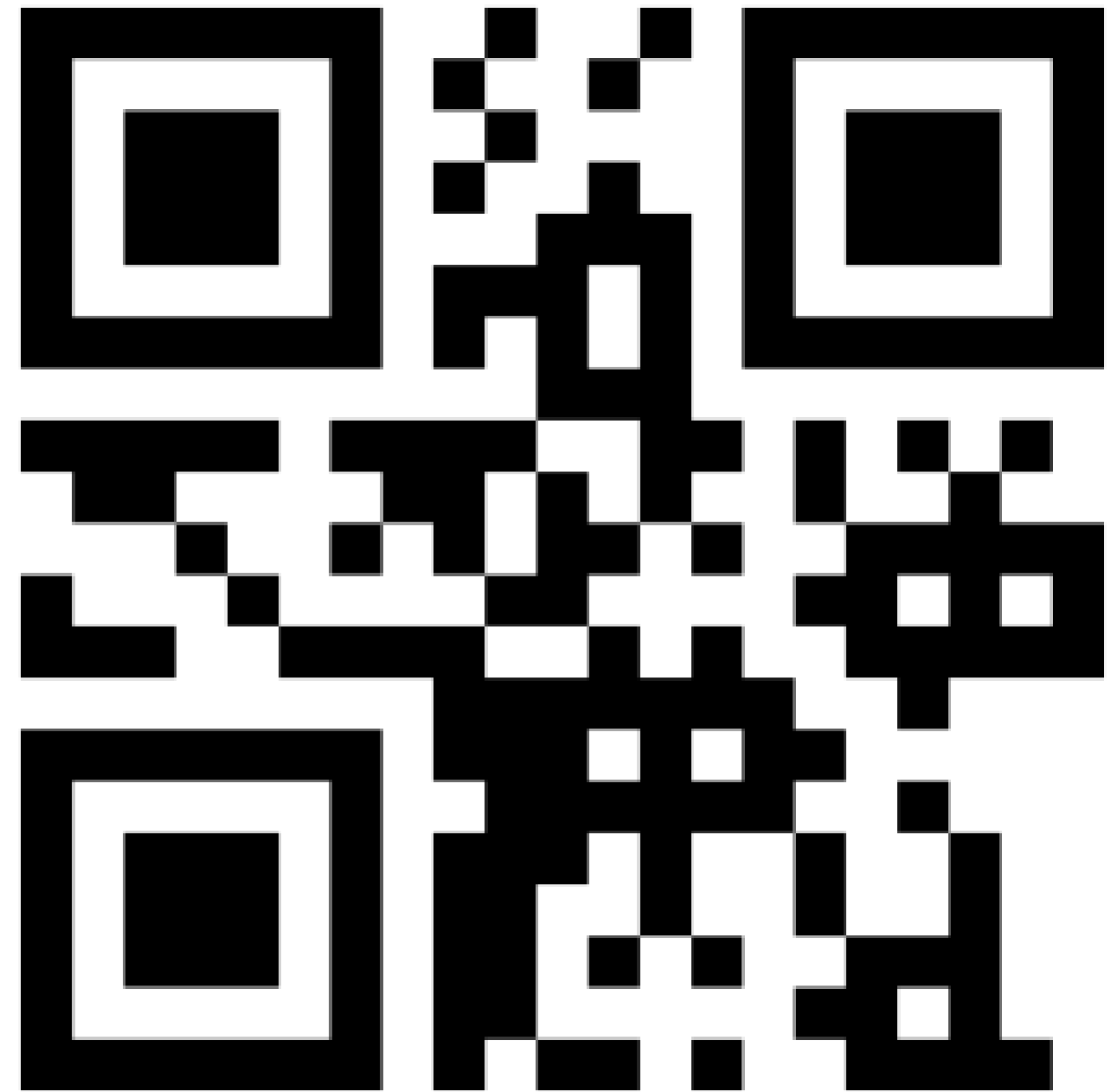
JJ_park_adventure.c

FEEDBACK?

**PLEASE LET ME KNOW ANY
FEEDBACK FROM TODAY'S
LECTURE!**

www.menti.com

Code: 4818 0273



WHAT DID WE LEARN TODAY?

ASSIGNMENT 1 IS RELEASED

Overview Live Stream:
Thursday 6:30pm
<https://youtu.be/QCgr8D>
HKDz8

REVIEW 1D ARRAYS

`highest_number.c`

WHAT IS A 2D ARRAY?

`print_2Darray.c`

2D ARRAYS: HARDER PROBLEMS

`JJ_park_adventure.c`

ANY QUESTIONS?

**DON'T FORGET YOU CAN
ALWAYS EMAIL US ON
CS1511@CSE.UNSW.EDU.AU
FOR ANY ADMIN QUESTIONS**

**PLEASE ASK IN THE FORUM
FOR CONTENT RELATED
QUESTIONS**

