

# Lecture 10

## 2D Array Practice with Tammy :)

COMP1511 Programming Fundamentals

# Intro

# Announcements

- Assignment 1 Livestream Recording (under Week 4)
- Lots of help Sessions* this week and next week! (+ some stages-specific help sessions!)
- Simple Snake lab this week to help with Assignment 1 (+ this lecture :D)



*Help Session timetable here: https://cgi.cse.unsw.edu.au/~cs1511/23T1/help-sessions/

# This Lecture...

Concepts (maybe) transferrable to assignment 1 -
Practice with a problem about 2D array of structs
(a.k.a. array of arrays of struct)

# Live Code (hopefully)

# Quick Revision

# Quick Revision

## Structs

1. *Define (above main)*

```c
struct coordinate {
    int x_coordinate;
    int y_coordinate;
};
```

2. *Declare*

```c
struct coordinate cood_point;
```

3. *Initialise*

```c
cood_point.x_coordinate = 3;
cood_point.y_coordinate = 5;
```



## Enums

1. *Define (above main)*

```c
enum icecream {Dulce, Vanilla, Choc, Pistachio, Strawberry, Mint}
```

2. *Declare* `enum icecream menu_choice;` 3. *Initialise* `menu_choice = Dulce;`

# Quick Revision

## 1D Arrays

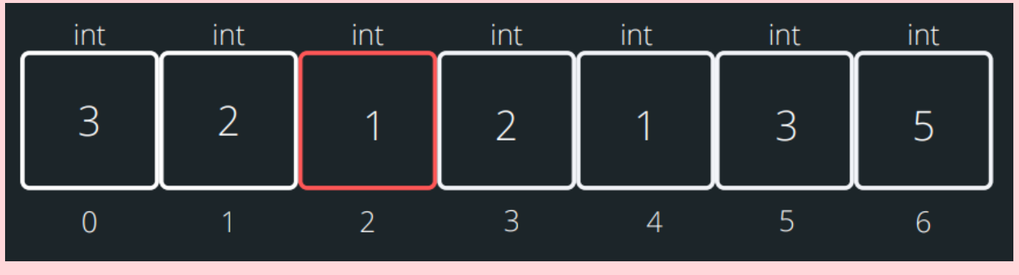
*1. Declare & Initialise*

```
int ice_cream_consum[7] = {3, 2, 1, 2, 1, 3, 5};
```

*2. Access an element*

```
ice_cream_consum[2]
```

*Visually:*




## 2D Arrays (Array of Arrays)

*1. Declare (can also initialise with a loop)*

```
int array[3][5];
```

*2. Access an element*

```
array[2][3];
```

*Visually:*

# Quick Revision

## 1D Array of Structs

*Assume a defined Struct like:*

```
struct coordinate {
    int x;
    int y;
};
```



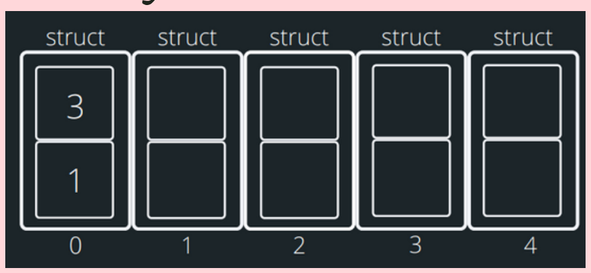
1. *Declare*

```
struct coordinate map[5];
```

2. *Initialise*

```
map[0].x = 3;
map[0].y = 1;
```

*Visually:*




## 2D Array of Structs (containing enums)

What we are going through today :)

# Practice Problem:



# Context

# Practice Problem!

## Bubble Tea Adventure

I have moved to a new home and am craving for bubble tea (as usual).

But I don't know this area very well - so we will go on an adventure, (navigating using WASD keys around a map) to look for a bubble tea (boba) store.

# Practice Problem!

## Bubble Tea Adventure

This problem has been broken down into 5 smaller tasks:

1. Get user input for initial details about the map - *home location (coordinates), boba shop location (coordinates).*
2. Update the map with these details.
3. Keep getting user input of 'w' (up), 'a' (left), 's' (down),'d' (right), update and print the updated map until I find the boba store.
4. [If time allows] Add code to get more user input (as a part of the initial details) to build a big gym (2x2) (so I can stay healthy whilst drinking more boba) - *gym location (starting coordinate).*
5. [If time allows] Allow user to give up before finding a boba store by pressing ctrl+ d.

# Some similarity to Assignment 1...

We have some starter code to work with, containing code to setup, including functions to:

- initialise_map
- print_map (and print_location - known as print_tile in assignment 1)



```
void initialise_map(struct location map[MAP_ROWS][MAP_COLUMNS]);
void print_map(struct location map[MAP_ROWS][MAP_COLUMNS]);
void print_location(struct location location, int place_print);
```

# Set Up

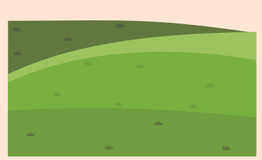
```
enum entity {
    PERSON,
    BOBA,
    FOOTPRINT_UP,
    FOOTPRINT_DOWN,
    FOOTPRINT_LEFT,
    FOOTPRINT_RIGHT,
    EMPTY
};
```

**EMPTY**



```
enum place_type {
    SHOP,
    GYM,
    HOME,
    UNDEVELOPED
};
```





# Set Up

```
enum entity {
    PERSON,
    BOBA,
    FOOTPRINT_UP,
    FOOTPRINT_DOWN,
    FOOTPRINT_LEFT,
    FOOTPRINT_RIGHT,
    EMPTY
};
```

```
enum place_type {
    SHOP,
    GYM,
    HOME,
    UNDEVELOPED
};
```

Examples of a struct location:




```
struct location {
    enum entity entity;
    enum place_type place;
};
```

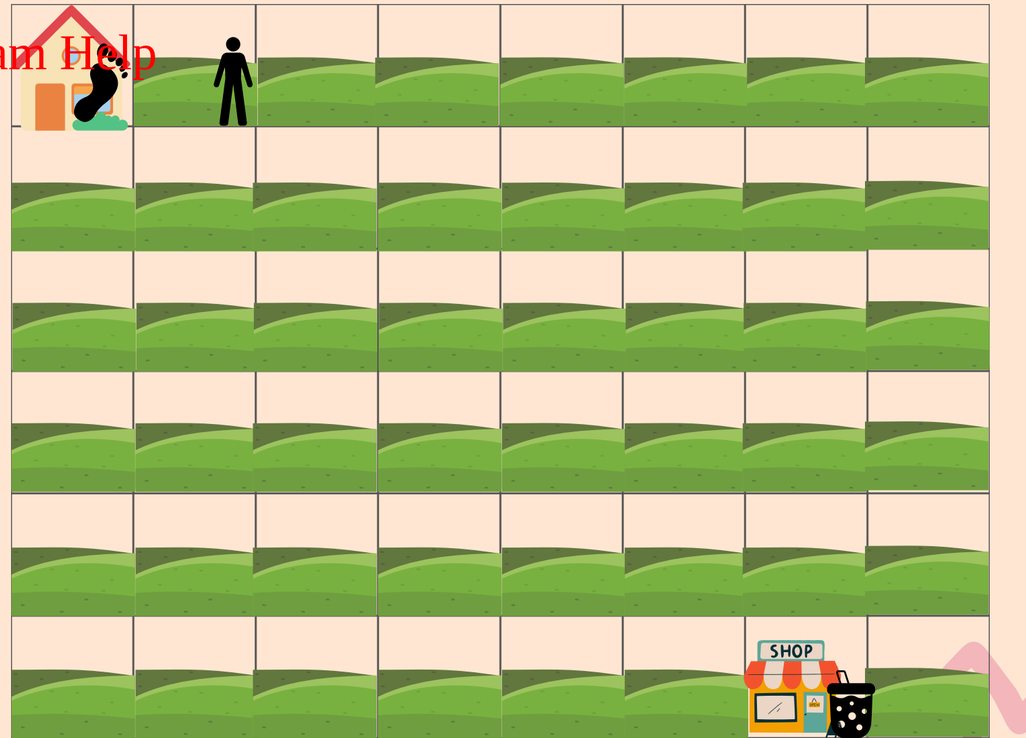
# Set Up

```
                                    6                8
struct location map[MAP_ROWS][MAP_COLUMNS];
```

Examples of a struct location:




```
struct location {
    enum entity entity;
    enum place_type place;
};
```




Every single cell on the map is a struct location!

# Set Up

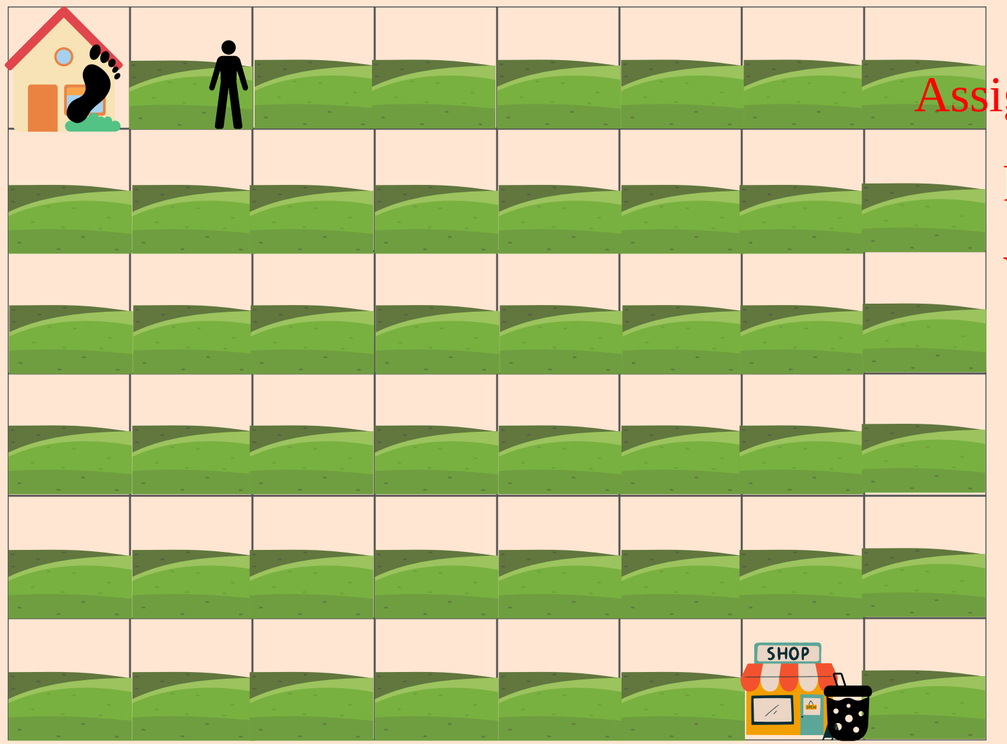
## The 2D Array of Structs

```c
struct location {
    enum entity entity;
    enum place_type place;
};
```

```c
struct location map[MAP_ROWS][MAP_COLUMNS];
```

6 8

Graphically:




How we may visualise it in relation to code:

| | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 |
|---|---|---|---|---|---|---|---|---|
| 0 | entity == FOOTPRINT place == HOME | entity == PERSON place == UNDEVELOPED | entity == EMPTY place == UNDEVELOPED | | | | | |
| 1 | | | | | | | | |
| 2 | | | | | | | | |
| 3 | | | | | | | | |
| 4 | | | | | | | | |
| 5 | | | | | | | entity == BOBA place == SHOP | |

**AND SO ON...**

# Set Up

## The 2D Array of Structs

```
struct location {
    enum entity entity;
    enum place_type place;
};
```

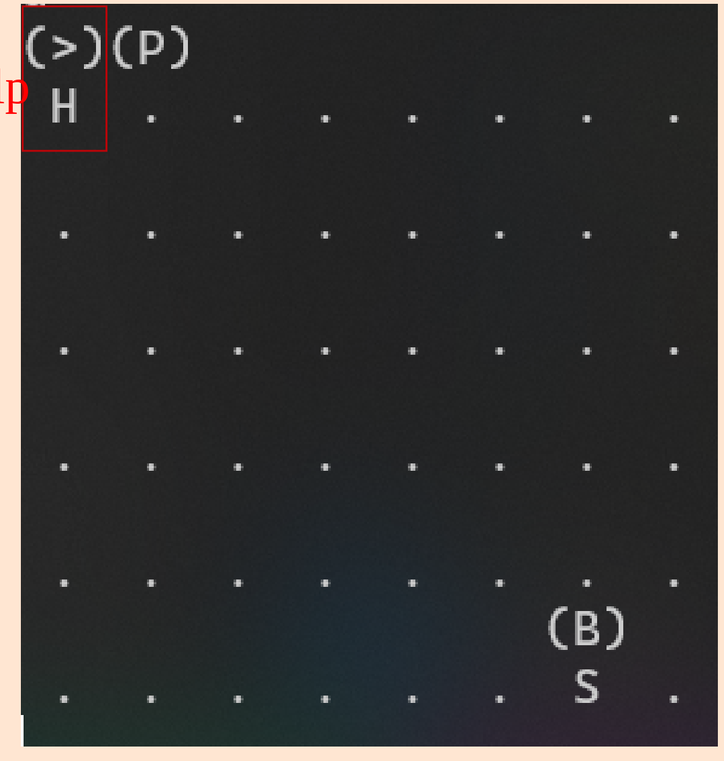
6 8

```
struct location map[MAP_ROWS][MAP_COLUMNS];
```

How we may visualise it in relation to code:

| | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 |
|---|---|---|---|---|---|---|---|---|
| 0 | entity == FOOTPRINT place == HOME | entity == PERSON place == UNDEVELOPED | entity == EMPTY place == UNDEVELOPED | | | | | |
| 1 | | | | | | | | |
| 2 | | | AND SO ON... | | | | | |
| 3 | | | | | | | | |
| 4 | | | | | | | | |
| 5 | | | | | | | entity == BOBA place == SHOP | |

How we show it on the terminal - the print_map function does this for us:




The circled part is an example of map[0][0]!

# Practice Problem:



# Coding Time!

# Practice Problem!

## Task #1



Get user input for initial details about the map - home location (coordinates), boba shop location (coordinates).

# Practice Problem!

## Task #2



If the inputs are valid, update the map with the boba shop and home location then print out the initial map.

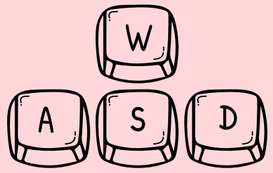
# Practice Problem!

## Task #3



Keep getting user input of 'w' (up), 'a' (left), 's' (down),'d' (right), update and print the updated map until I find the boba store.

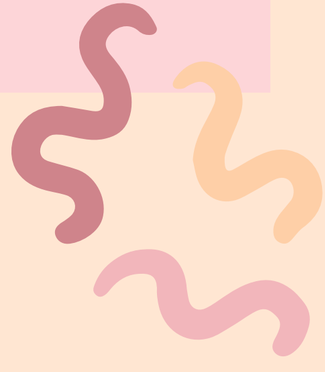
Once that is working, add code to leave footprints where you have explored!



# Break Time!

# Code Style





# *Some* of the Things in the Assignment 1 Style Rubric

**Let's look at these in the context of the code we wrote!**

- Functions
  - 2 ways you can go about this depending on whether you feel confident about functions
- #defines for magic numbers
- Comments
- Line length

NOTE: Style is marked manually in your assignment 1 but the 1511 style checker can help you pick up on some smaller issues. (Make sure you are also following the 1511 style guide!)

# Practice Problem:



# Back to Coding!

# Practice Problem!

## Task #4



Add code to get more user input (as a part of the initial details) to build a big gym (2x2) (so I can stay healthy whilst drinking more boba) - gym location (starting coordinate).

# Practice Problem!

## Task #5



Allow user to give up before finding a boba store by pressing ctrl+ d.

# Feedback

## (pretty please with a cherry on top)

This is my first ever 1511 lecture, I would really appreciate any feedback to help me improve my teaching <3

https://www.menti.com/aligwybon37r

# Summary

## Assignment 1 Livestream

Recording under week 4 on course website!

## 2D Array of Structs



Building on from what we learnt about enums, structs, arrays, 2D arrays.

## Practice Problem

Bubble Tea Adventure!

# If you have any questions

## Course Related:

Course Forum + Help Sessions!



## Admin Related:

cs1511@unsw.edu.au



**Thank you everyone :)**