

DPST1091 / CPTG1391

# Introduction to Programming

## Week 4 – Lecture 1

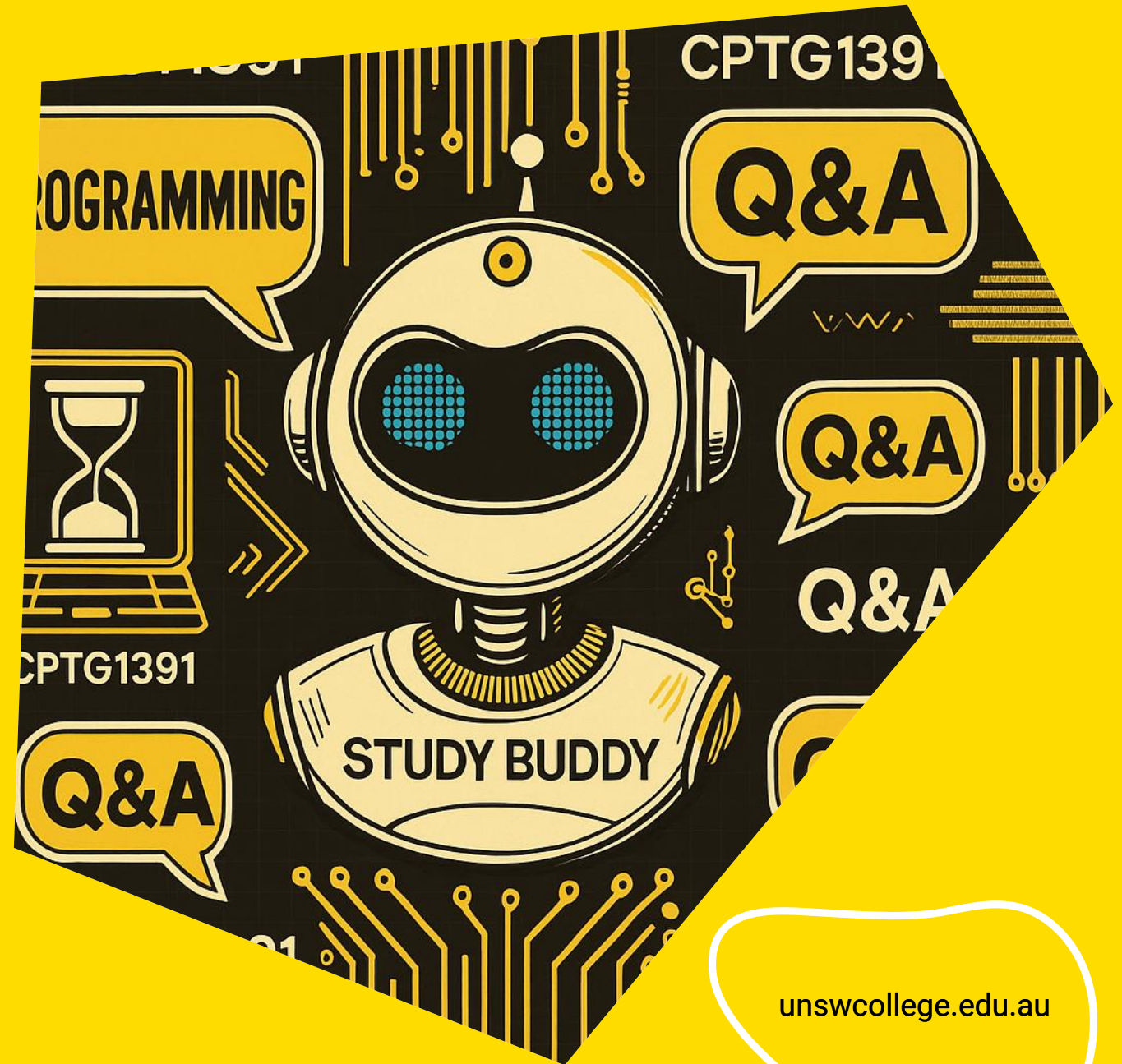
Lecturer and Course Convener:

**Dr Pantea Aria**



UNSW  
College

# 2D arrays



[unswcollege.edu.au](http://unswcollege.edu.au)

# Agenda

- **Last lecture**
  - 1D arrays

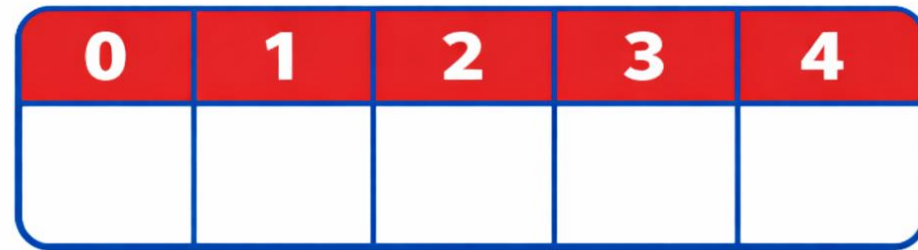
- **Today**
  - **arrays of structs**
  - **2D arrays**

# Arrays recap

- An **array** is a way to store **many values** together.
- All values in an array must be the **same type**.
- We use **one name** to refer to the whole array.
- We can **access** any element at any time using **its position**.

# Array Declaration

```
int quiz_scores[5];
```



# Declare and initialise

```
int quiz_scores[5] = {12, 15, 18, 20, 17};
```

```
// initialise all elements to 0  
int quiz_scores[5] = {};
```

```
// this will create a 4-element array  
int ages[] = {18, 19, 20, 21};
```

```
// this Will create a 14-element array,  
// with the first 7 elements initialised and  
// the remaining 7 set to 0  
int age[14] = {3, 2, 1, 2, 1, 3, 5};
```

# Accessing the elements

```
int quiz_scores[5] = {12, 15, 18, 20, 17};
```



0	1	2	3	4
12	15	18	20	17

```
int first_score = quiz_scores[0];
```

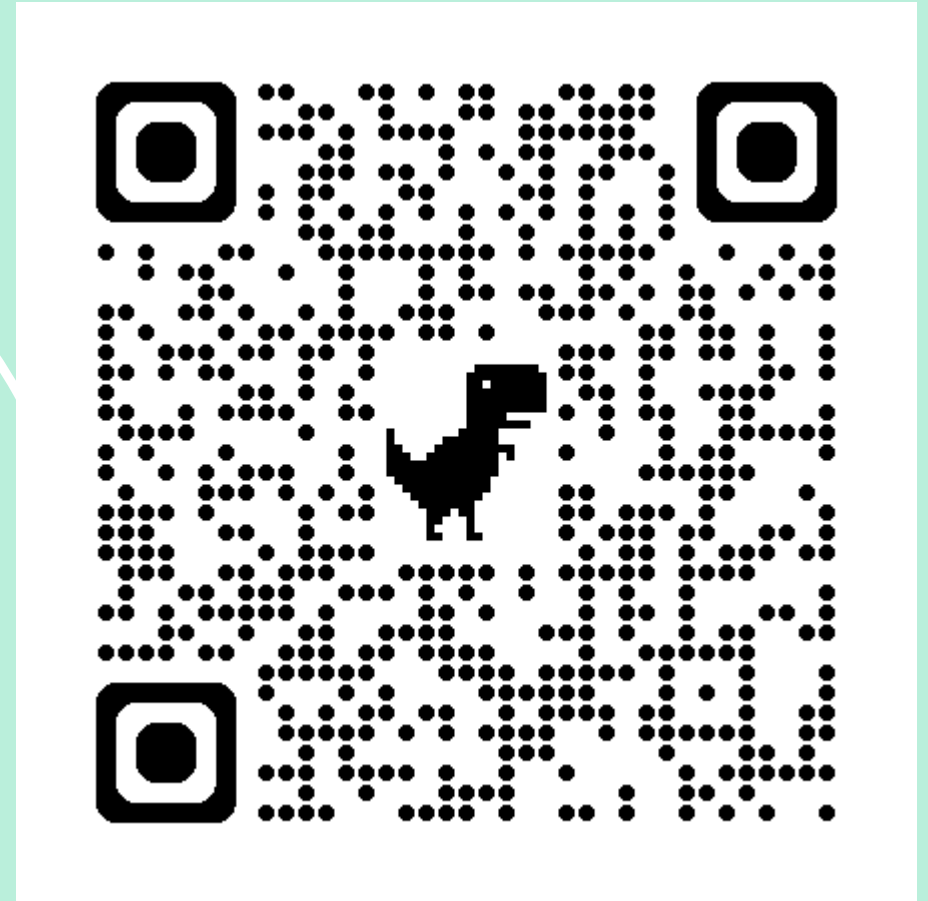
first\_score

12

# Demo

→array\_recap.c

Live lecture code is written for teaching, not perfection.  
It may include extra comments and may not always follow  
ideal coding style



# Arrays and Functions

Arrays can be **passed** into functions just like other variables.

When **passing an array**, the function must also be told **how many elements** the array contains.

This allows the function to safely work with **arrays of different sizes**.

```
// Function that works with an array of integers of any length  
void display_array(int length, int numbers[]);
```

# Example

```
void display_array(int length, int numbers[]);
```

```
int main(void) {  
    int temperatures[] = {18, 21, 25, 19};  
    int scores[] = {100, 75};
```

```
    display_array(4, temperatures);  
    display_array(2, scores);
```

```
    return 0;
```

```
}
```

```
void display_array(int length, int numbers[]) {
```

```
    int i = 0;
```

```
    while i < length) {
```

```
        printf("%d ", numbers[i]);
```

```
        i++;
```

```
    }
```

```
}
```

# Important Behaviour of Arrays in Functions

When an array is passed to a function, **the entire array is not copied.**

The function works with the **original array** created in the caller function.

**Any changes made inside the function affect the array outside the function.**

This behaviour will be explained in more depth in **later weeks.**

# Initialising an Array Using a Function

- A **function** can be used to **fill an array** with values.
- The **array is created in main** and **passed** into helper functions.

```
#define SIZE 5

int main(void) {
    int values[SIZE];

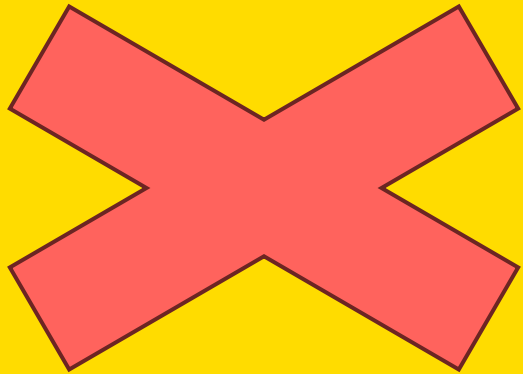
    read_values(SIZE, values);
    display_array(SIZE, values);

    return 0;
}

void read_values(int length, int array[]) {
    int i = 0;
    while (i < length) {
        scanf("%d", &array[i]);
        i++;
    }
}
```

array is not  
a copy of  
the **values**  
array.

# Returning Arrays from Functions



- Trying to **return an entire array** from a function **does not work**.
- Although the code **may compile**, it leads to serious **runtime problems**.
- We will explore why this happens later in the course.

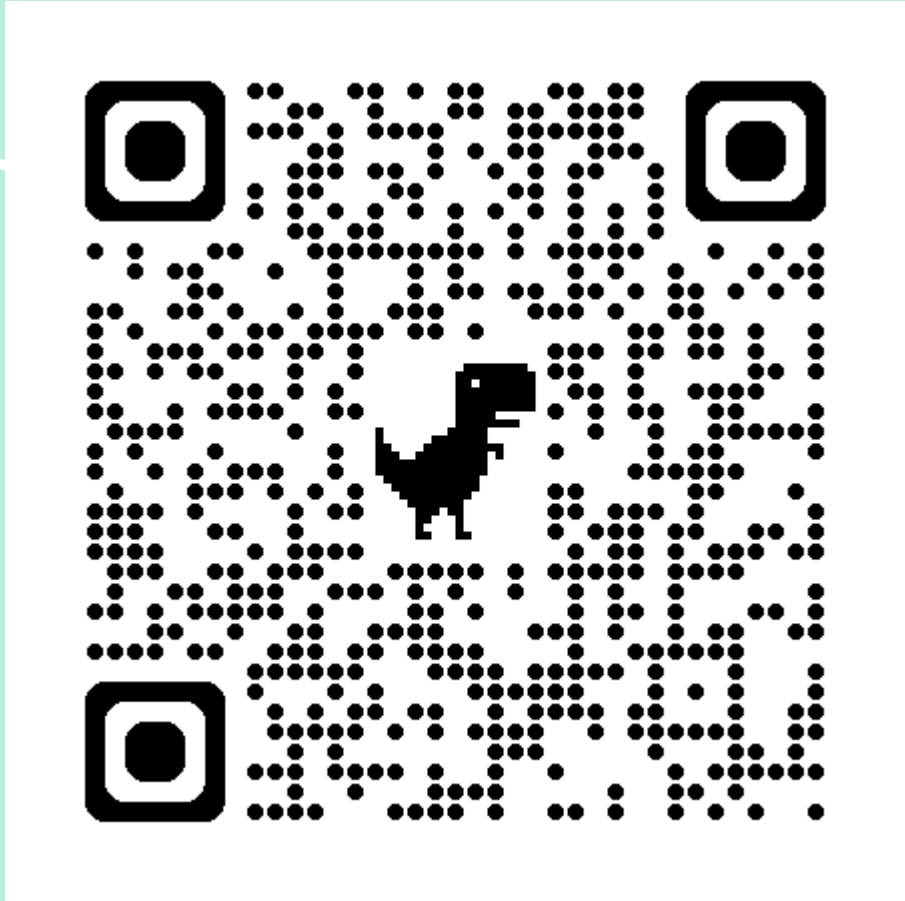
```
// This does NOT work
int[] read_values(void) {
    int array[SIZE];

    for (int i = 0; i < SIZE; i++) {
        scanf("%d", &array[i]);
    }

    return array;    // X invalid
}
```

# Demo

→array\_function\_recap.c

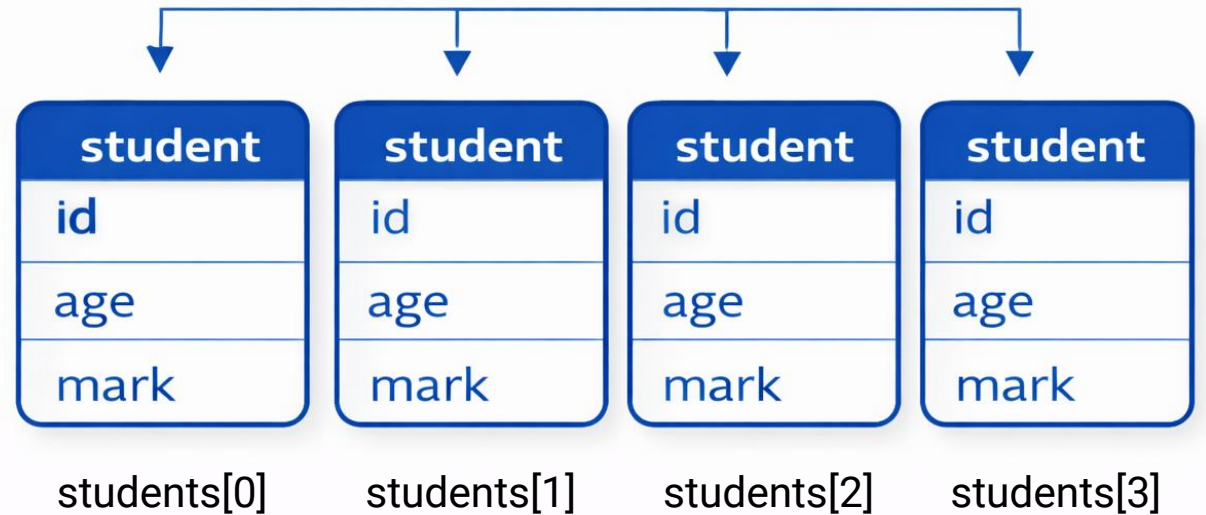


Live lecture code is written for teaching, not perfection.  
It may include extra comments and may not always follow  
ideal coding style

# Array of Struct

```
struct student {  
    int id,  
    int age,  
    double mark;  
};
```

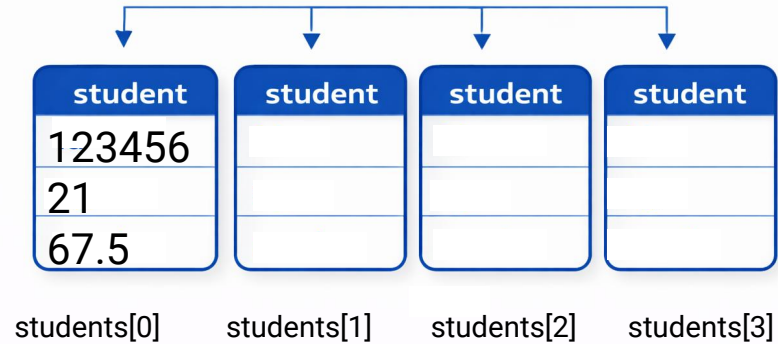
```
struct student students[4];
```



# Accessing the elements of an array of structures

```
struct student {  
    int id,  
    int age,  
    double mark;  
};
```

```
struct student students[4];
```

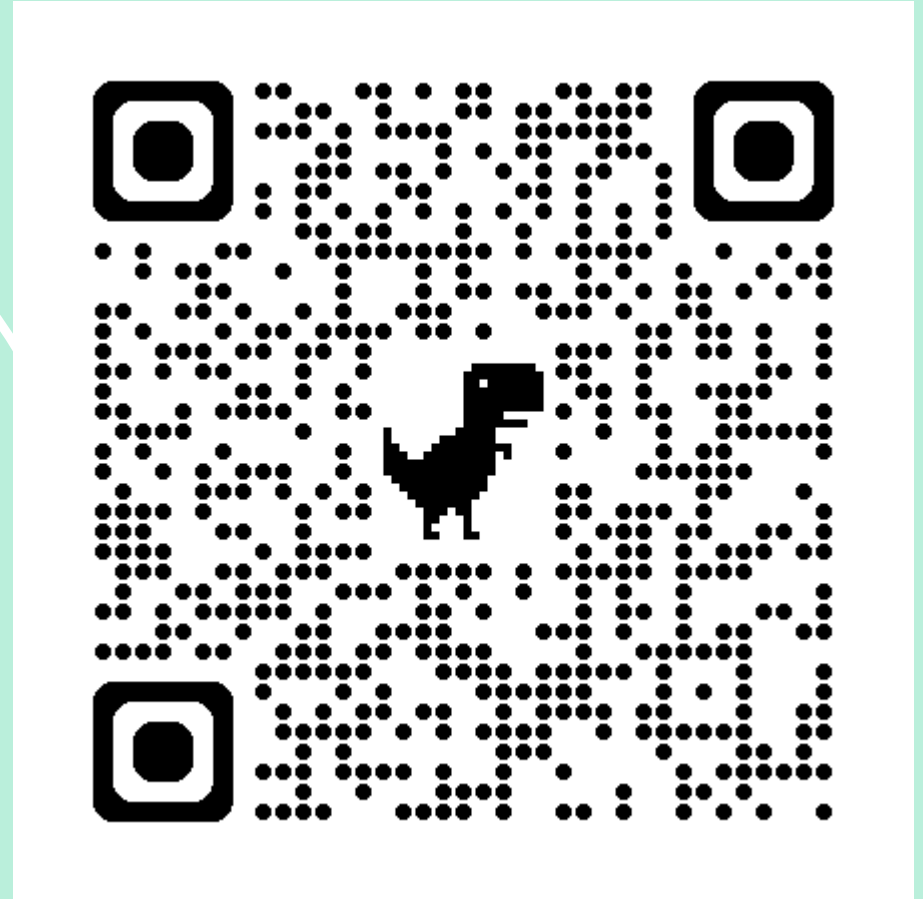


```
students[0].id = 123456;  
students[0].age = 21;  
students[0].mark = 67.5;
```

# Demo

→array\_of\_struct.c

Live lecture code is written for teaching, not perfection.  
It may include extra comments and may not always follow  
ideal coding style



# IT'S BREAK TIME!

```
#include <stdio.h>
#define ON_BREAK 1
int main(){
    // Time for a 10 minute break! Switch to PARTY_MODE
    #define PARTY_MODE ON_BREAK
    if {PARTY_MODE == ON_BREAK) ;
        print("Program will resume in 10 minutes...");
        sleep(600); // Take a break
        exit(0);
}
```

**10 MINUTES BREAK!**

Relax... We'll be back soon!

# 2 Dimensional Arrays

## arrays of arrays

It stores data in **rows and columns**, like a **table or grid**.  
How to declare?

```
<type> <identifier>[<rows>][<cols>;
```

```
// Declaring a 2D array  
int marks[3][4];
```

This means:

3 rows

4 columns

Total elements =  $3 \times 4 = 12$

You can picture it like this:

```
[] [] [] []  
[] [] [] []  
[] [] [] []
```

# Initialising a two-dimensional array

```
// Initialising a 2D array
int marks[3][4] = {
    {10, 12, 14, 16},
    {11, 13, 15, 17},
    {18, 19, 20, 21}
};
```

Each inner {} is one row.

	col 0	col 1	col 2	col 3
row 0	10	12	14	16
row 1	11	13	15	17
row 2	18	19	20	21

marks

# How to traverse the array

A 2D array has  
two dimensions:

- rows
- columns

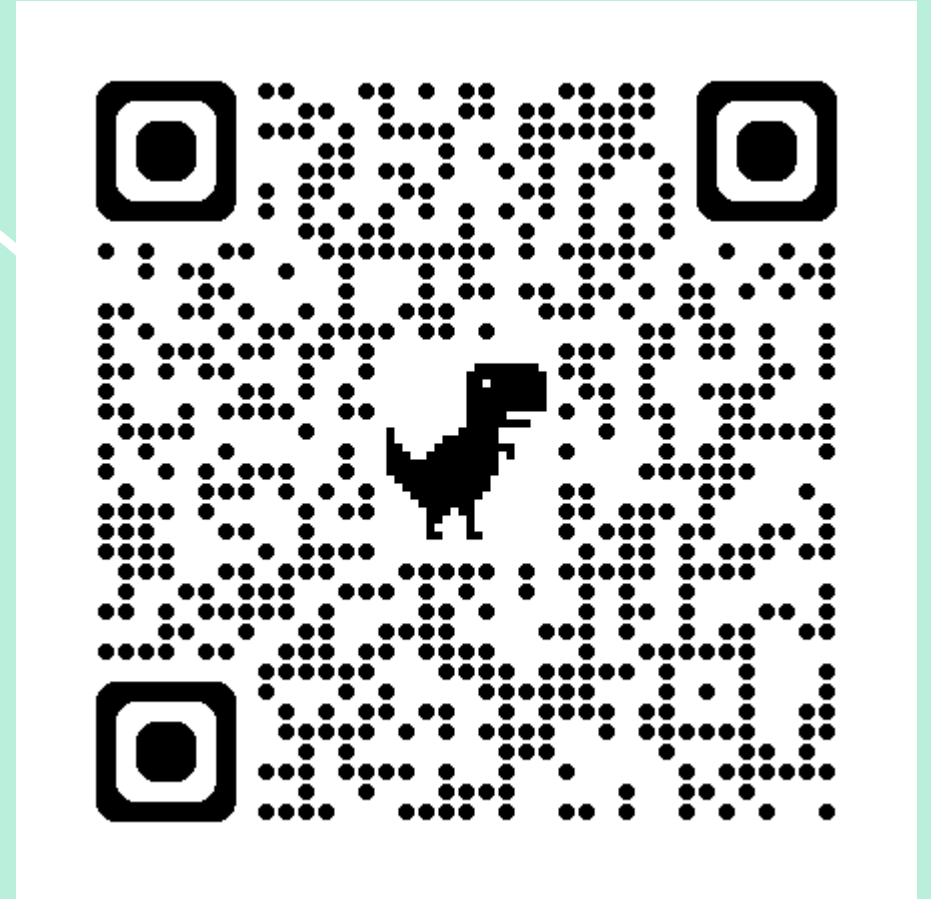
So, we need:

- one loop to move through the **rows**
- one loop to move through the **columns**

```
int row = 0;
// outer loop for rows
while (row < 3) {
    int col = 0;
    // inner loop for columns
    while (col < 4) {
        printf("%d ", marks[row][col]);
        col++;
    }
    printf("\n");
    row++;
}
```

# Demo

- `2Darrays_input_elements`
- `2Darrays_print_elements`
- `2Darrays_sum_elements`
- `2Darrays_sum_diagonal`



Live lecture code is written for teaching, not perfection.  
It may include extra comments and may not always follow  
ideal coding style

# Voice of the Student

Anonymous ongoing feedback  
Anything you wanted to share with me



26T1 Voice of the Student



[26T1 Voice of the Student – Fill out form](#)

**See you soon ...**