An array of arrays, 2D
LAST WEEK... 

IN WEEK 3... 

- 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
THIS LECTURE...

TODAY...

- Recap basic arrays
- Array of structs
- Array of arrays
Live lecture code can be found here:

HTTPS://CGI.CSE.UNSW.EDU.AU/~CS1511/23T1/LIVE/WEEK04/
Assignment 1 will be released after this lecture

CS Defence - based on tower defence games

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

IN-PERSON AND LIVESTREAM

- 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:
  - Tuesday 7th March 2:00pm
  - In-person K17 Level 1 Room 113 (Seminar Room)
  - Live: https://youtube.com/live/bAygNaTepa8?feature=share
RECAP OF ARRAYS

Remember that arrays:

- are a collection all of the same type
- are 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
So let’s say we have this declared and initialised:

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

This is what it looks like visually:

```
   int  int  int  int  int  int  int
  3    2    1    2    1    3    5
  0    1    2    3    4    5    6
```

This array holds 7 integers

Note that indexing starts at 0
RECAP OF ARRAYS

- You can access any element of the array by referencing its index.
- Note, that indexes start from 0.
- Trying to access an index that does not exist, will result in an error.

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

If I wanted the third element of the array:

The index would be 2, so to access it:

```c
ice_cream_consum[2]
```
RECAP OF ARRAYS

AN EXAMPLE PROBLEM

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

`lowest_number.c`

Problem: A user is asked to enter 10 numbers. We will then go through these numbers and find the sum of the even numbers only.

`sum_even.c`
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?
The struct for a coordinate point:
```c
struct coordinate {
    int x;
    int y;
};
```

An array of structs declared:
```c
struct coordinate map[5];
```

An array of structs visually:
```c
map[0].x = 3;
map[0].y = 1;
```
ACCESSING AN ELEMENT INSIDE ARRAY OF ARRAYS

An array of arrays is basically a grid. To declare an array of arrays:

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

To access an element now you will need to:

```c
array[2][3];
```
Think of the problem last week where we tracked ice-cream consumption for a week. What if I want to do this for a month (a week at a time)?

```c
int ice_cream[4][7];
```
Do you remember when we printed out a grid of numbers in Week 2 with our while inside a while?

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

How can we transfer this knowledge to print out an array of arrays?
int array[3][4];
int row = 0;
while (row <= 3){
    int col = 0;
    while (col <= 4){
        printf("%d", array[row][col]);
        col++;
    }
    printf("\n");
    row++;
}
TRANSFER THIS TO AN ARRAY:

FIRST RUN AROUND THE SUN: OUTSIDE WHILE ROW = 0 INSIDE WHILE COL = 1

```c
int array[3][4];
int row = 0;
while (row <= 3){
    int col = 0;
    while (col <= 4){
        printf("%d", array[row][col]);
        col++;
    }
    printf("\n");
    row++;
}
```
TRANSFER THIS TO AN ARRAY:

FIRST RUN AROUND THE SUN:
OUTSIDE WHILE ROW = 0
INSIDE WHILE COL = 2

```c
int array[3][4];
int row = 0;
while (row <= 3){
    int col = 0;
    while (col <= 4){
        printf("%d", array[row][col]);
        col++;
    }
    printf("\n");
    row++;
}
```
TRANSFER THIS TO AN ARRAY:

FIRST RUN AROUND THE SUN:
OUTSIDE WHILE ROW = 0
INSIDE WHILE COL = 3

```c
int array[3][4];
int row = 0;
while (row <= 3){
    int col = 0;
    while (col <= 4){
        printf("%d", array[row][col]);
        col++;
    }
    printf("\n");
    row++;
}
```
TRANSFER THIS TO AN ARRAY:

SECOND RUN AROUND THE SUN: OUTSIDE WHILE ROW = 1
INSIDE WHILE COL = 0

```c
int array[3][4];
int row = 0;
while (row <= 3){
    int col = 0;
    while (col <= 4){
        printf("%d", array[row][col]);
        col++;
    }
    printf("\n");
    row++;
}
```
int array[3][4];
int row = 0;
while (row <= 3){
    int col = 0;
    while (col <= 4){
        printf("%d", array[row][col]);
        col++;
    }
    printf("\n");
    row++;
}
TRANSFER THIS TO AN ARRAY:

SECOND RUN AROUND THE SUN: OUTSIDE WHILE ROW = 1 INSIDE WHILE COL = 2

```c
int array[3][4];
int row = 0;
while (row <= 3){
    int col = 0;
    while (col <= 4){
        printf("%d", array[row][col]);
        col++;
    }
    printf("\n");
    row++;
}
```
TRANSFER THIS TO AN ARRAY:

SECOND RUN AROUND THE SUN: OUTSIDE WHILE ROW = 1
INSIDE WHILE COL = 3

```c
int array[3][4];
int row = 0;
while (row <= 3){
    int col = 0;
    while (col <= 4){
        printf("%d", array[row][col]);
        col++;
    }
    printf("\n");
    row++;
}
```
TRANSFER THIS TO AN ARRAY:

THIRD RUN AROUND THE SUN:
OUTSIDE WHILE ROW = 2
INSIDE WHILE COL = 0

```c
int array[3][4];
int row = 0;
while (row <= 3){
    int col = 0;
    while (col <= 4){
        printf("%d", array[row][col]);
        col++;
    }
    printf("\n");
    row++;
}
```
TRANSFER THIS TO AN ARRAY:

THIRD RUN AROUND THE SUN: OUTSIDE WHILE ROW = 2 INSIDE WHILE COL = 1

```c
int array[3][4];
int row = 0;
while (row <= 3){
    int col = 0;
    while (col <= 4){
        printf("%d", array[row][col]);
        col++;
    }
    printf("\n");
    row++;
}
```
TRANSFER THIS TO AN ARRAY:

THIRD RUN AROUND THE SUN: OUTSIDE WHILE ROW = 2 INSIDE WHILE COL = 2

```c
int array[3][4];
int row = 0;
while (row <= 3){
    int col = 0;
    while (col <= 4){
        printf("%d", array[row][col]);
        col++;
    }
    printf("\n");
    row++;
}
```
TRANSFER THIS TO AN ARRAY:

THIRD RUN AROUND THE SUN:
OUTSIDE WHILE ROW = 2
INSIDE WHILE COL = 3

```c
int array[3][4];
int row = 0;
while (row <= 3){
    int col = 0;
    while (col <= 4){
        printf("%d", array[row][col]);
        col++;
    }
    printf("\n");
    row++;
}
```
Let’s try our hand at using a 2D array to solve a problem!

Declare and create a 3 x 3 two-dimensional array of integer numbers with the numbers read in from the user. Then loop through the two-dimensional array, printing out the values in the first row followed by those in the second row and so on.

- Now loop through the array to count the number of even numbers in the 2D array

2D_Arrays.c
I value your feedback and use it to pace the lectures and improve your overall learning experience. If you have any feedback from today’s lecture, please follow the link below. Please remember to keep your feedback constructive, so I can action it and improve the learning experience.

https://www.menti.com/albo2hw32oxo
WHAT DID WE LEARN TODAY?

ASSIGNMENT 1 IS RELEASED
LIVESTREAM on Tuesday 2:00pm
In-person: K17, Room 113 Seminar

RECAP 1D ARRAYS
lowest_number.c
sum_even.c
array of structs:
dogs.c

AN ARRAY OR ARRAYS (2D)
2D_array.c
REACH OUT

CONTENT RELATED QUESTIONS
Check out the forum

ADMIN QUESTIONS
cs1511@unsw.edu.au