

Tute 04

COMP1511 22T3

Jack Robbers

content

- functions
- scanning in loops
- arrays

assignment 0

- what did we learn

functions

- live coding
- `make_colour()`
- `get_main_colour()`
- `invert_colours()`

scanning in loops

```
int main(void) {  
    int number;  
    while (scanf("%d", &number) == 1) {  
        printf("%d\n", number)  
    }  
}
```

what happens when you type in:

- a number
- a letter
- a few numbers with spaces between them
- nothing

arrays

- arrays store things of the same type sequentially
- in groups, write the following functions
- swap who is holding the pen after each instruction
- if you get time, add a main function

instructions

Odd Only - `void odd_only(int array[SIZE])`

example input - `odd_only([1, 2, 3, 4, 5, -10]);` (SIZE is 6 in this case)

- Create a while loop which loops through every element of the array.
- Write an if statement which adds 1 to each even value. Do this within the while loop.
- Write another while loop which goes through the array with a different iterator (i.e. if you used i last time, use j)
- Print out the values in the array.

Copy Array - `void copy_array(double from[SIZE], double to[SIZE])`

example input - `copy_array([3.1415, 2.71828, 1.4142], [0.0, 0.0, 0.0]);` (SIZE is 3 in this case)

- Create a while loop that 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 that prints out all the elements of the second array.

Largest Character - `char largest_character(char array[SIZE])`

example input - `printf("%c \n", largest_character(['C', 'O', 'M', 'P', 'I', 'S', 'I', 'I']));` (SIZE is 8 in this case)

- Create a character variable called largest_character, equal to the first character of the array.
- Create a while loop to loop through the character array.
- Create an if statement to check if the current character has a higher ascii value than "largest_character"
- Return the largest character you've found.

