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

```c
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)

1. Create a while loop which loops through every element of the array.
2. Write an if statement which adds 1 to each even value. Do this within the while loop.
3. Write another while loop which goes through the array with a different iterator (i.e. if you used i last time, use j)
4. 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)

1. Create a while loop that loops through every element of the first array.
2. Copy the elements of the first array into the second array (leave 0's at the end)
3. 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', '1', '5', '1', '1']));` (SIZE is 8 in this case)

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

odd_only

```c
void odd_only(int array[SIZE]) {
    // 1. Create a while loop which loops through every element of the array.
    int i = 0;
    while (i < SIZE) {
        // 2. Write an if statement which adds 1 to each even value. Do this within the while loop.
        if (array[i] % 2 == 0) {
            array[i] = array[i] + 1;
        }
        i++;
    }

    // 3. Write another while loop which goes through the array with a different iterator
    // (i.e. if you used i last time, use j)
    int j = 0;
    while (j < SIZE) {
        // 4. Print out the values in the array.
        printf("%d\n", array[j])
        j++
    }
}
```

copy_array

```
void copy_array(double from[SIZE], double to[SIZE]) {
    // 1. Create a while loop that loops through every element of the first array.
    int i = 0
    while (i < SIZE) {
        // 2. Copy the elements of the first array into the second array (leave 0's at the end)
        to[i] = from[i];
        i++;
    }

    // 3. Create a while loop that prints out all the elements of the second array.
    int j = 0;
    // I chose to put them all on the same line in the format: [1.0, 2.0, ...]
    // this is just showing another way of printing an array
    printf("[")
    while (j < SIZE) {
        printf("%lf", to[i]);
        j++;
        if (j != SIZE) {
            printf(", ")
        }
    }
    // end the line
    printf("]\n");
}
```

largest_character

```c
char largest_character(char array[SIZE]) {
    // 1. Create a character variable called largest, equal to the first character of the array.
    char largest = array[0];

    // 2. Create a while loop to loop through the character array.
    int i = 0;
    while (i < SIZE) {
        // 3. Create an if statement to check if the current character
        // has a higher ascii value than "largest_character"
        if (array[i] > largest) {
            largest = array[i]
        }

        i++
    }

    // 4. Return the largest character you've found.
    return largest
}
```