Week 3

- Functions
- Passing variables into functions
- Returning values from functions

What was annoying last week?

- Math had to happen in place
- No way to reuse code without copy/pasting
- Huge main Method

What do we want to do?

- Create functions to reuse code
- Write slimmer main methods
- Explain what we do through method names

Our goals today?

- Create functions for the Math we did last week
- Do some recursive magic
- Understand most of C weirdness regarding functions

A function in C

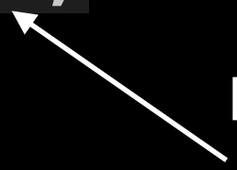
```
return_type function_name (parameter_list) {
    body
}
```

Return type

- If we want to return a result after something is calculated, we can return it
- Write which type is returned, e.g. int, float or other
- Each path in the function must return a value if stated
- Write void if you don’t want to return anything

```c
int return2(){
   return 2;
}

int main(){
    int two = return2();
    return 0;
}
```



Returns Exit code
0 is ok
Most other numbers are not

Parameter List

- All parameters which are passed into a function
- Separated by comma

```c
int add(int a, int b){
  return a+b;
}

int main(){
    int two = add(2,5);
    printf("%i\n",two);
    return 0;
}
```

Collatz

- Famous conjecture
- Given any integer number
- If divisible by 2 -> Divide by 2
- If not -> Multiply by 3, add 1
- Every number supposedly converges to 1

Will yield a warning

```c
int div2(int a){
  int tmp = a / 2;
  if(tmp == 1){
    return 1;
  }
  if(tmp % 2 == 0){
    return div2(tmp);
  }else{
    return mul3plus1(tmp);
  }
}
int mul3plus1(int a){
  int tmp = 3 * a + 1;
  if(tmp % 2 == 0){
    return div2(tmp);
  }else{
    return mul3plus1(tmp);
  }
}


int main(){
    int two = mul3plus1(257);
    printf("%i\n",two);
    return 0;
}
```

Yes this code is bad

Make better.

```
~/Desktop
→ gcc test.c
test.c:18:12: warning: implicit declaration of function 'mul3plus1' is invalid in C99 [-Wimplicit-function-declaration]
    return mul3plus1(tmp);
           ^
1 warning generated.
```

```c
int div2(int a){
  if(tmp % 2 == 0){
    return div2(tmp);
  }else{
    return mul3plus1(tmp);
  }
}
```

What is mul3plus1 ?

```c
int mul3plus1(int a){
  if(tmp % 2 == 0){
    return div2(tmp);
  }else{
    return mul3plus1(tmp);
  }
}
```

Function declaration

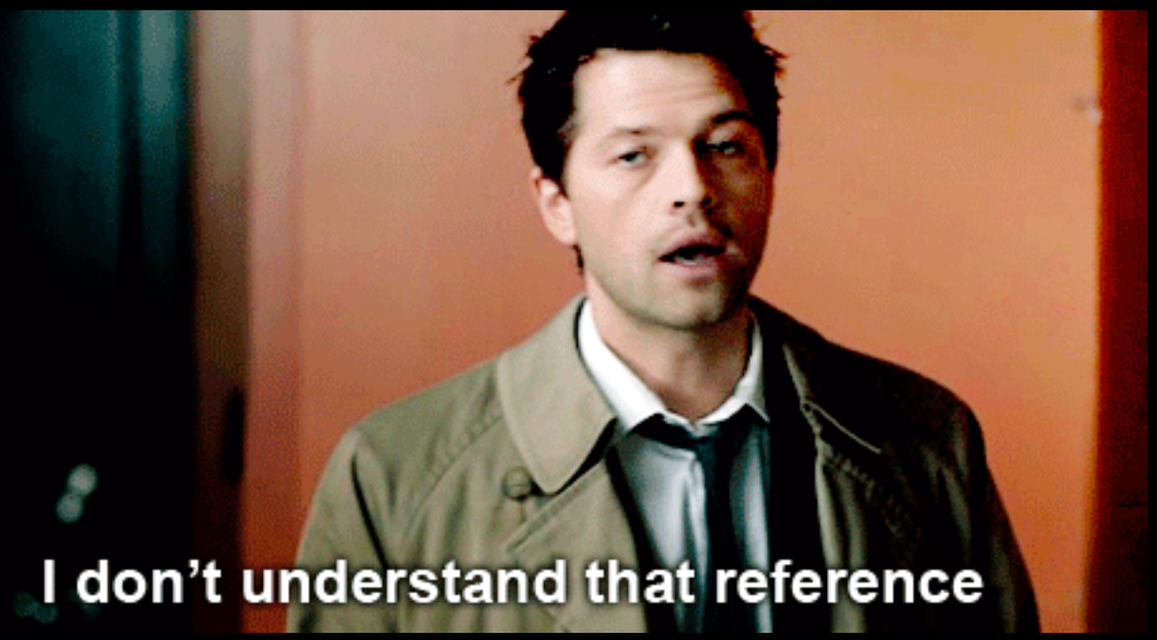
- C reads files top to bottom
- Any called function must be known beforehand

```c
int div2(int a);
int mul3plus1(int a);
```

At the top of the file

What happens if we call a function which has no body?

```
➜ gcc test.c
Undefined symbols for architecture x86_64:
  "_a", referenced from:
      _main in test-9cf8b0.o
ld: symbol(s) not found for architecture x86_64
clang: error: linker command failed with exit code 1 (use -v to see invocation)
```



–gcc

Exercises

- Write an add function to add two ints and one to add two floats

- Write a better function to calculate the Collatz series and print each number out

- Write a function that calculates the sum from all numbers between x and y

More Excercises

- Write two add functions, one for ints and one for floats
- Write functions to set/unset bits in an uint8_t (Highly relevant for Embedded Systems)
- Write a function that returns whether an input int is prime and which prime number it is
- Write a function that converts lowercase characters to upper case