

A large red square with a white border, centered on a white background. Inside the square, the text "Break and Continue" is written in white.

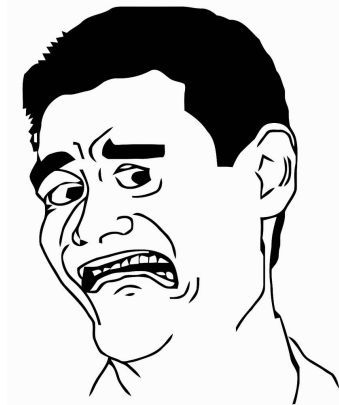
**Break and  
Continue**

# Coding Practices

- “Good Code”
  - Easy to understand
  - Easy to read
  - If you had to give it to someone who didn’t know what you were working on, they could figure it out in <5 minutes



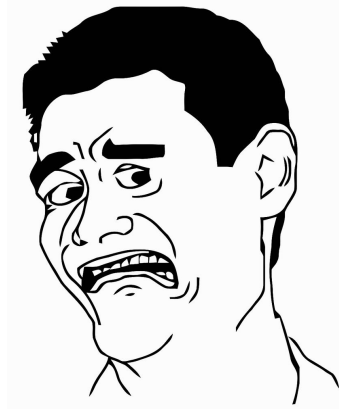
# Coding Practices



- “Bad Code”
  - Opposite of good code
  - Still compiles, but is extremely difficult to figure out what it does
- International Obfuscated C Code Contest
  - <http://www.ioccc.org/>

# Break / Continue

- Provide ways to stop a loop BESIDES the terminating condition
- Usually considered “bad code”
  - Loops “should” have only one way to exit (the termination condition)
  - BUT there are some situations where it comes in handy...



# Break

- Causes the loop to end IMMEDIATELY whenever the command *break*; is executed

```
int i;  
for (i = 0; i < 10; i++) {  
    if (i == 5) {  
        break;  
    }  
}
```



# Prime Numbers



- Numbers that cannot be divided by anything other than themselves and 1
  - Ex: 1, 3, 5, 7, 11, 13, 17, 23, etc...
- Example: Writing a code that can determine if a number is prime or not can use a *break* statement
  - There are better ways to do this...but this is one example

# Continue

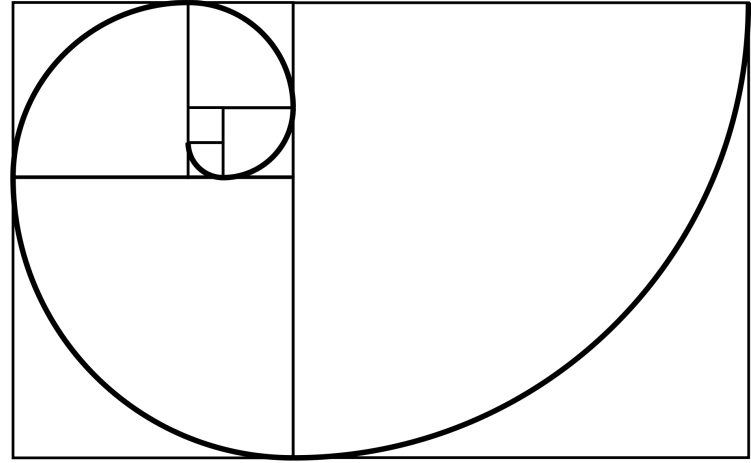
- Causes the remainder of a single repetition to be skipped

```
int i;  
for (i = 0; i < 10; i++) {  
    if (i == 5) {  
        continue;  
    }  
    printf("%d\n", i);  
}
```



# Fibonacci Numbers

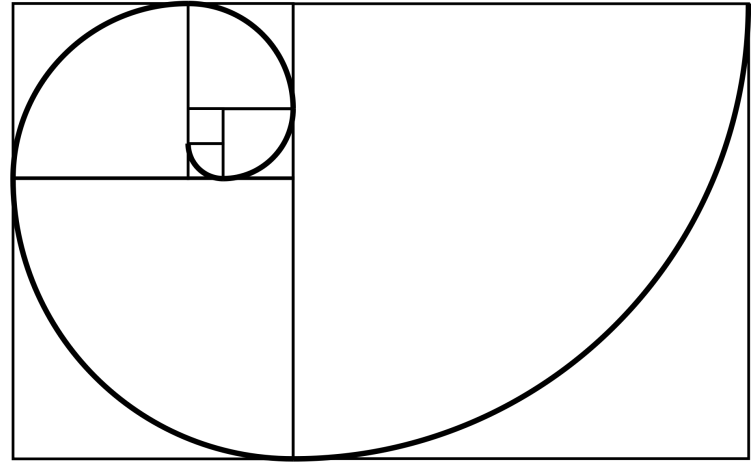
- 1, 1, 2, 3, 5, 8, 13...
- What comes next / What is the pattern?





# Fibonacci Numbers

- 1, 1, 2, 3, 5, 8, 13...
- Creating a fibonacci sequence of length  $n$  can use a break statement
  - There are better and more efficient ways to do this...but this is one way



# Continue Coding Challenge

You are in charge of counting down a NASA liftoff  
→ but there is a specific script that you have to follow. You have to say

*10, 9, 8, Main engines start, 6, 5, 4, 3, 2, 1, Liftoff*

, exactly in that order. Make a for loop that counts down from 10-0 and prints out your exact script, using *continue* statements.

