THINK IN C

Session 3

FUNCTION

A reusable piece of code

Your code is a collection of functions

SCOPE

Local

SCOPE

Global

SYNTAX

```
<return data type> <function name> (<arguments>) {
    /* code */
    return <return value>;
}
```

VOID

WHY MAIN IS WRITTEN THE WAY IT IS

```
int main(int argc, char const *argv[]) {
    /* code */
    return 0;
}
```

ORDER OF PASSING

```
int Function(int a, int b, char k ){
    return (a/b);
}
int main (int argc, char const *argv[]) {
    int a, b;
    b = Function(b, a, ‘A’);
    return 0;
}
```

USING LIBRARY FUNCTIONS

#include <.h>

#include “.h”

What is a .h file?

Why a .h file?

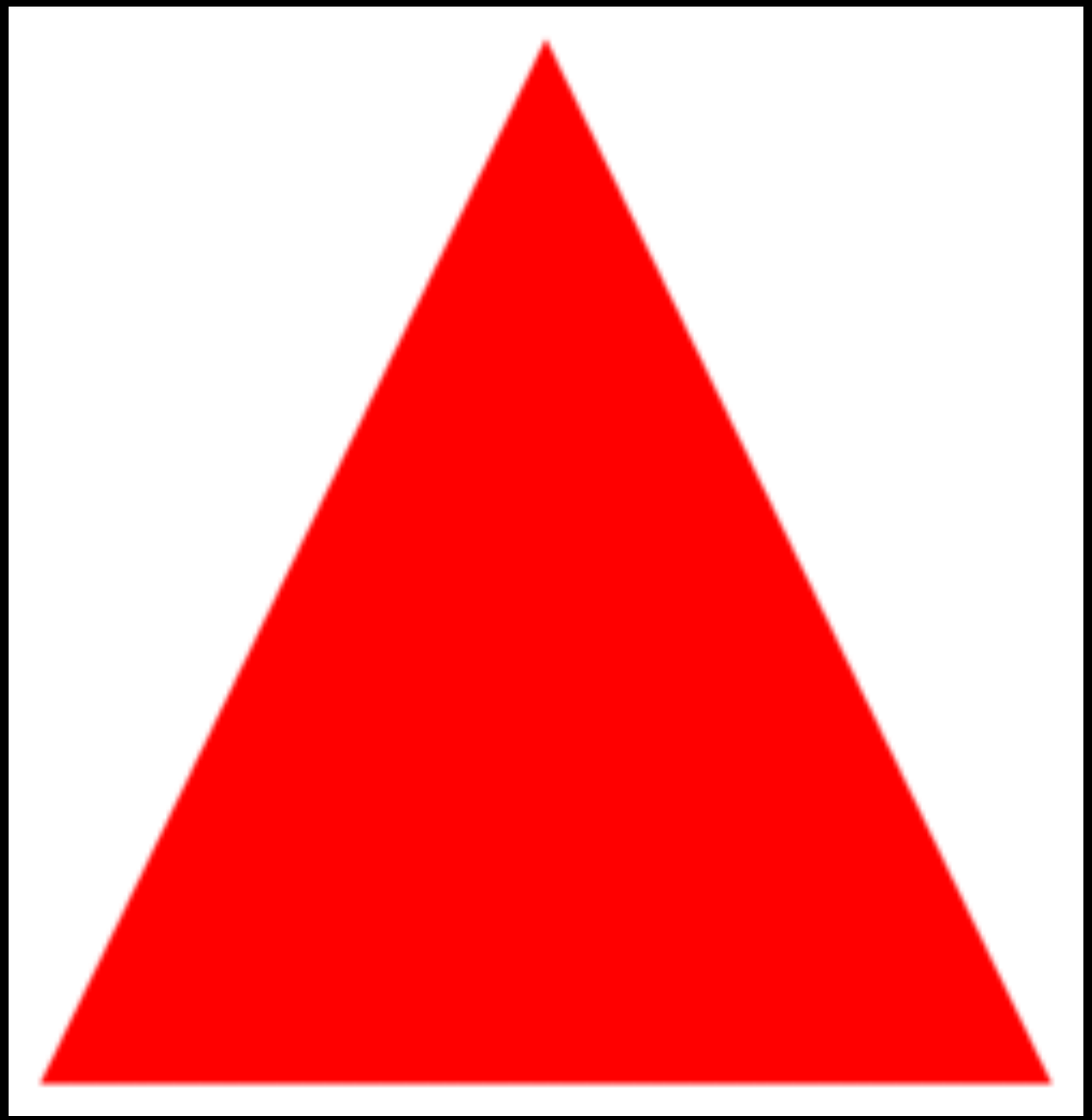
RETURN TYPE

Why Return?

Return What?

RECURSION

RECURSION IN MATH



SYNTAX

```
<return data type> <function name> (<arguments>){
    /*code*/
    <function name>(<arg>);
    /*code*/
   return <return value>;
}
```

EXAMPLE

```
int rec (int a) {
    if (a<1){
        return 1;
    }
    else{
    return (a * rec( a-1));
    }
}
```

EXAMPLE

rec(5) return(5 times rec(4),

which return (4 times rec(3),

which return (3 times rec(2),

which return (2 times rec(1),

which returns (1)))))

LET'S CODE

rec(5) return(5 times rec(4),

which return (4 times rec(3),

which return (3 times rec(2),

which return (2 times rec(1),

which returns (1)))))

Can you code?

- Fibonacci series
- Prime or not?
- Prime or not? (using Recursion)