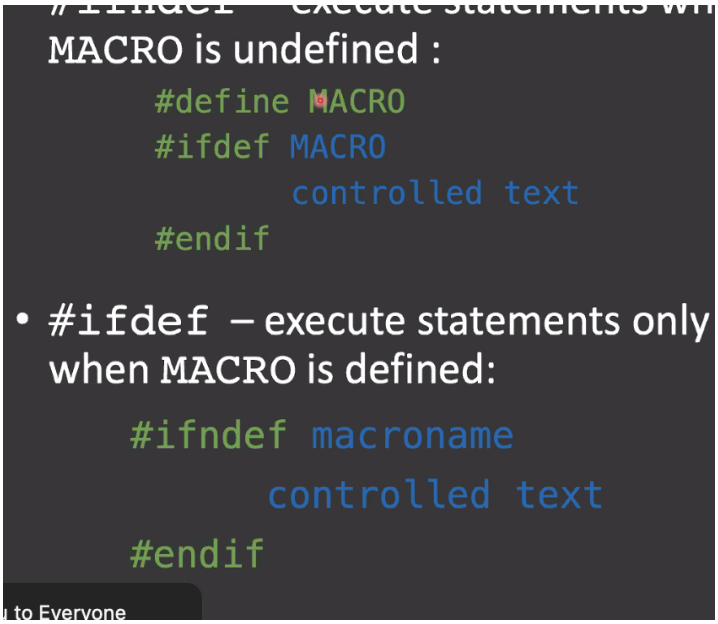
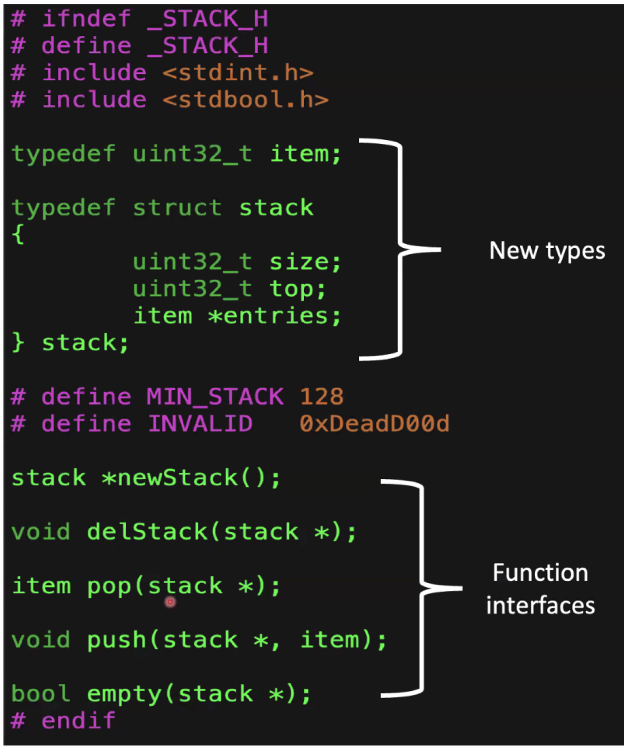
I. Function Prototypes
II. What is #include
    A. Preprocessor directive
        1. Happens before compilation
        2. Processes all #
    B. Paste code of the given file into current file
    C. Used to include functions defined in other libraries
    D. #define
        1. Directive to the preprocessor
        2. Replace all instances of word
    E. Conditional directives
        1. Conditional statements in the prepocessor



III. Header Files
    A. Only have things shared between source code files
        1. Function, Macro declarations
        2. Data structures
        3. Global Variables
        4. #include directive
    B. Uses extension .h
    C. Guard the headers with Conditional preprocessors

1.




```
# ifndef _STACK_H
# define _STACK_H
# include <stdint.h>
# include <stdbool.h>

typedef uint32_t item;

typedef struct stack
{
        uint32_t size;
        uint32_t top;
        item *entries;
} stack;

# define MIN_STACK 128
# define INVALID   0xDeadD00d

stack *newStack();

void delStack(stack *);

item pop(stack *);

void push(stack *, item);

bool empty(stack *);
# endif
```

D. Standard header files - stdio.h, inttypes.h, time.h, stdbool.h, ctype.h, math.h

IV. Extern

A. Allows C to access variables and functions

B.

```
#include "extern.h"

int counter = 42; // Global counter definition.

void decrement(void) {
  counter--;
  return;
}

void increment(void) {
  counter++;
  return;
}
```

```
#ifndef __EXTERN_H__
#define __EXTERN_H__

extern int counter; // Counter declaration in extern.c.

void decrement(void); // Function prototype for decrement().

void increment(void); // Function prototype for increment().

#endif
```

V. Static

A. Declared inside and outside function

B. Value of variable

VI. Recursion

A. Calls itself

VII.