

# Objectives - Thu 1/31/2019

- Quiz 2: GDB
- HW02 lessons
  - Outline of a solution (to help with HW04)
- Variadic arguments
- How a program is born
  - editor → preprocessor → compiler → linker → executable
- Preprocessor
  - #define, #include, ... – all just simple text substitution
  - #define macros – best practices

# Mistakes on HW02

fputc(45, stdout)  
~~45~~

<del>48</del>	✓	'0'
<del>97</del>	✓	'a'
<del>87</del>	✓	'a'-10

Duplication (DRY rule)

```
switch (digit-value) {  
    case '0':  
        .....
```

```
}
```

Sprawling code → Use paragraphs

# Tips

Organize code into "paragraphs".

Choose variable names wisely.

(What if your code was on the exam?)

int max = ...;

int maximum = ...;

int 111111 = ..., 111111 = ...;

X 48

X 45

X '0'

X '-'

## Variadic functions

va\_arg et al are  
macros (like functions with  
special capabilities)

---

va\_arg can only take args  
in order, only one time.

---

Not an array.

Must call with correct type.

# Variadic functions

\$ man va\_arg

va\_start (more\_args format)

A va\_list is not an array

— One time\*

— In order

— In that function\*

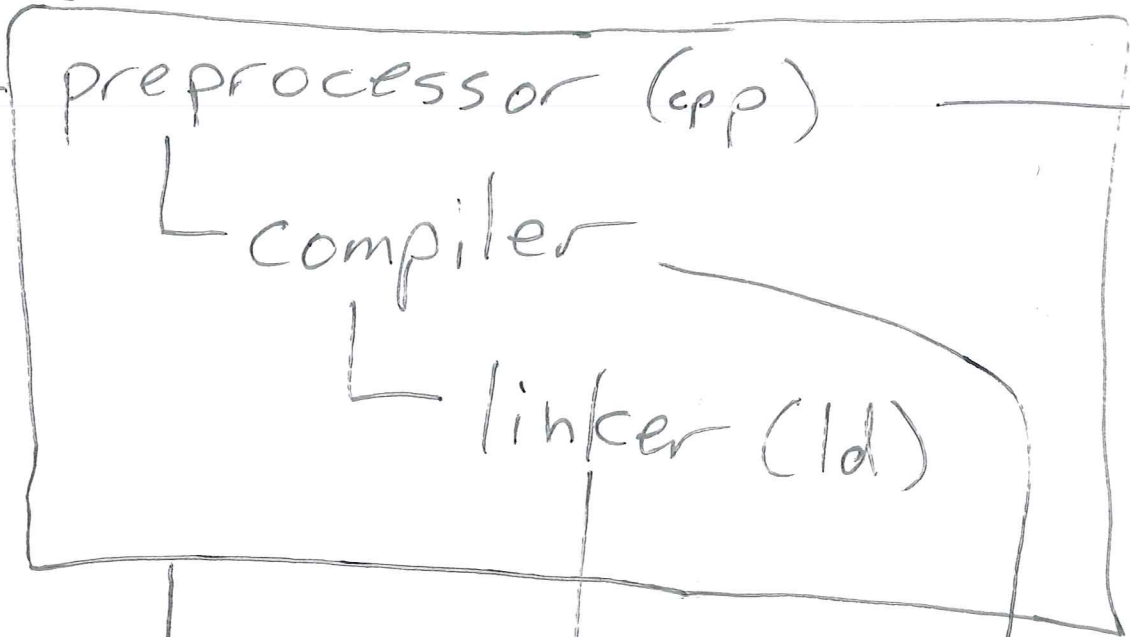
\* Workarounds are yucky

Birth of a program

Source code (vim)

gcc -o test\_mintf mintf.c test\_mintf.c

mintf.c test\_mintf.c



→ // → ""  
/\* → ""  
#define  
#include  
.....

→ binary executable

mintf.c → mintf.o  
test\_mintf.c → test\_mintf.o

mintf.o test\_mintf.o → test\_mintf