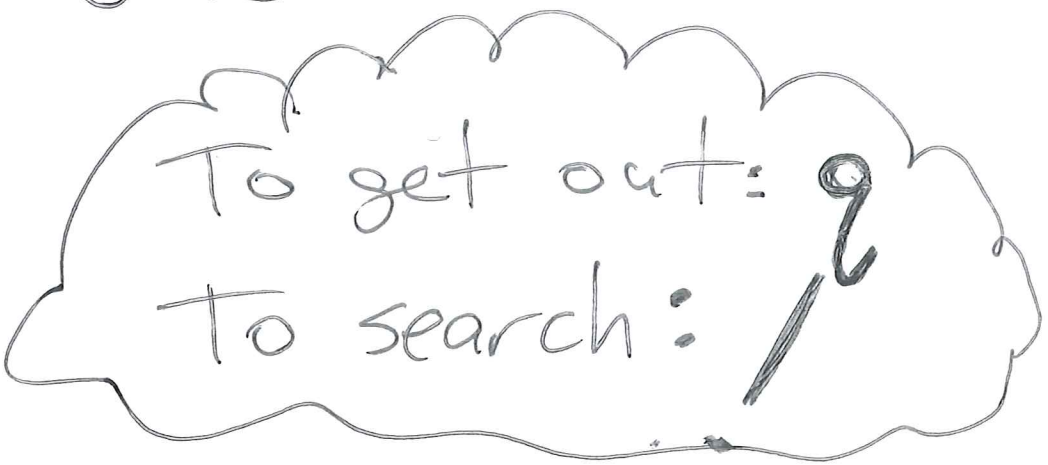
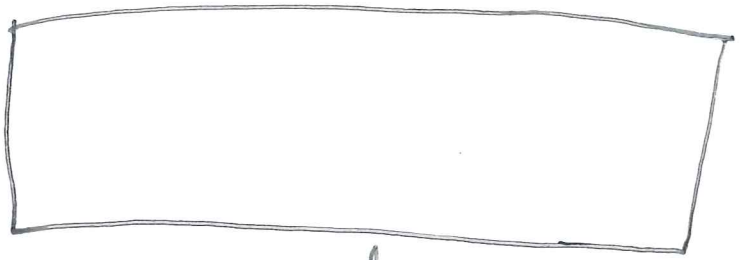
Objectives - Thu 3/21/2019

- Files
  - fopen(...)
  - fwrite(...)
  - fread(...)
  - feof(...)
  - strerror(...)

Man pages

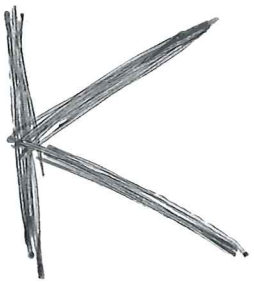
bash: man





command, e.g. ls
library function e.g. fopen
header file

Vim: fopen(----)

Press 

FILE* fp = fopen("a.txt", "w");
file pointer
open a file
filename
mode
"r"
"w"
"a"
read
write
append

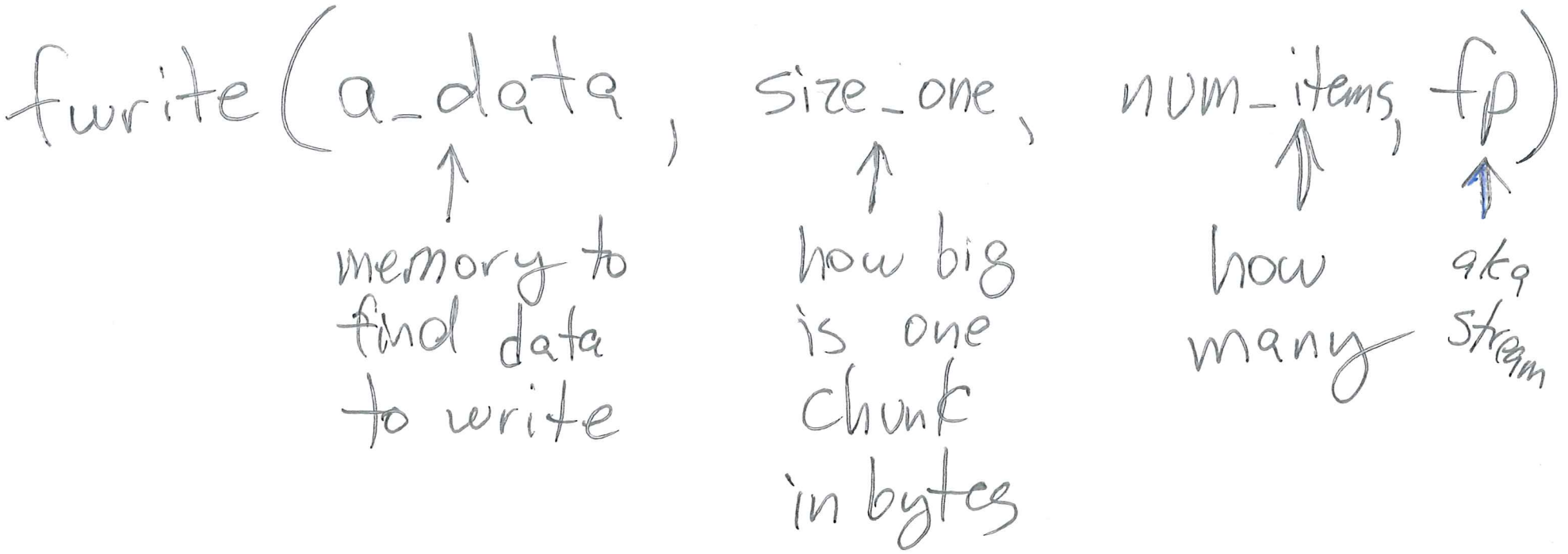
Don't use an assignment in a condition

```
if( fp = fopen(...) ) {
```

Boo!!!

```
}
```

```
while((c = fgetc(...)) != EOF)
```



Copies bytes from memory to disk.

feof(fp) returns (true) non-zero once fgetc(fp) (or equivalent) has tried and failed to read past the end of the file.

NOT just saying you've read the last character.

EOF is returned by fgetc to indicate end of file.

Use feof(fp),
not EOF to
check for end of file.

Streams	mode
FILE* fp	(depends on arg to fopen(...))
stdout	"w"
stderr	"w"
stdin	"r"

All can be passed to fwrite, fread, etc. if mode is right

fprintf(stderr, ~~~)

fprintf(stdout, ~~~)

fprintf(fp, ~~~)

$ same for fputc(~~~);

char: -128 to 127
(signed)

unsigned char: 0 to 255

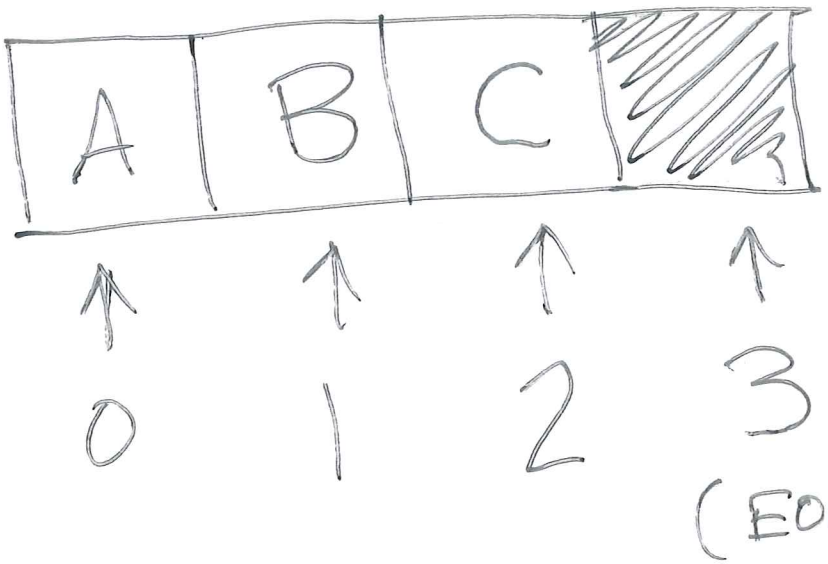
int: INT_MIN to INT_MAX
<<< -1 >>> 255

EOF -1

ftell( ) Where are we?

fseek( ) Go to ____.
(place in file)

a.txt



EOF is not a character in your file.

ftell( ) gives the idx to be read next

INIT

while (CONDITION) {

BODY....

NEXT

}

---

for (INIT ; CONDITION ; NEXT) {

BODY

}