Files - fopen, fclose

To open and close files for reading and writing.

```
#include <stdio.h>
...
FILE *in = fopen("inputfile", "r");     // Return NULL if error
FILE *out = fopen("outputfile", "w");  // Creates the file if needed
...
fclose(in);
fclose(out);
```

Files - fread, fwrite

Attempt to read `n` 1-byte items from `in` and write them in buffer at `buf`. Return number read.

```
int num_read = fread(buf, 1, n, in);
```

If `num_read < n` then end-of-file reached.

```
int num_written = fwrite(buf, 1, n, out);
```

`fwrite` reads `n` 1-byte items from `buf` and writes them to `out`.

Useful pattern

To read a file to the end, chunk by chunk, and do something with each.

```
unsigned char buf[BUF_BYTES];
int bytes_read  = fread(buf,1,BUF_BYTES,in);
while (bytes_read > 0) {
    // Do something with buf[0 ... bytes_read - 1]
    bytes_read  = fread(buf,1,BUF_BYTES,in);
}
```

If `bytes_read > 0` then something was read. If `bytes_read` is 0 then no more to read.

Example - Duplicate a file

```
#include <stdio.h>

static void dup(char *src_name, char *dst_name) {
    const int BUF_LEN = 512;
    FILE *src = fopen(src_name, "r");
    if (src != NULL) {
        FILE *dst = fopen(dst_name, "w");  // Creates new or overwrites old
        if (dst != NULL) {
            unsigned char buf[BUF_LEN];
            int bytes_read = fread(buf, 1, BUF_LEN, src);
            while (bytes_read > 0) {
                fwrite(buf, 1, bytes_read, dst);
                bytes_read = fread(buf, 1, BUF_LEN, src);
            }
            fclose(dst);
        }
        fclose(src);
    }
}
```