recover

TODO

- open memory card file
- find beginning of JPEG
- open a new JPEG
- write 512 bytes until new JPEG is found
- detect end of file

TODO

- open memory card file
  - `fopen`
- find beginning of JPEG
- open a new JPEG
- write 512 bytes until new JPEG is found
- detect end of file

TODO

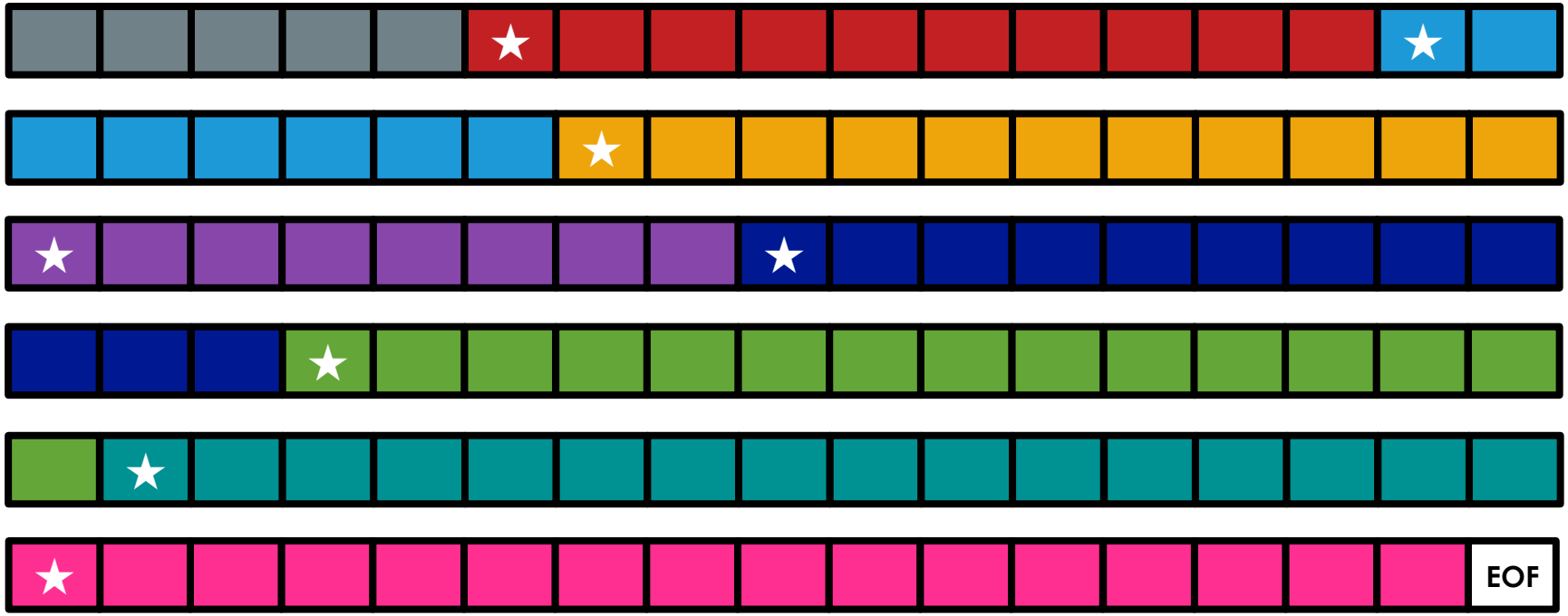
- ☑ open memory card file
- ☐ find beginning of JPEG
- ☐ open a new JPEG
- ☐ write 512 bytes until new JPEG is found
- ☐ detect end of file

JPEGs

- JPEGs are just sequences of bytes
- each JPEG starts with a distinct header:
  - first 3 bytes: `0xff 0xd8 0xff`
  - last byte: `0xe0, 0xe1, 0xe2, ... 0xef`
- stored side-by-side on the memory card
- each block is 512 bytes

memory card

Each ☐ represents 512 bytes

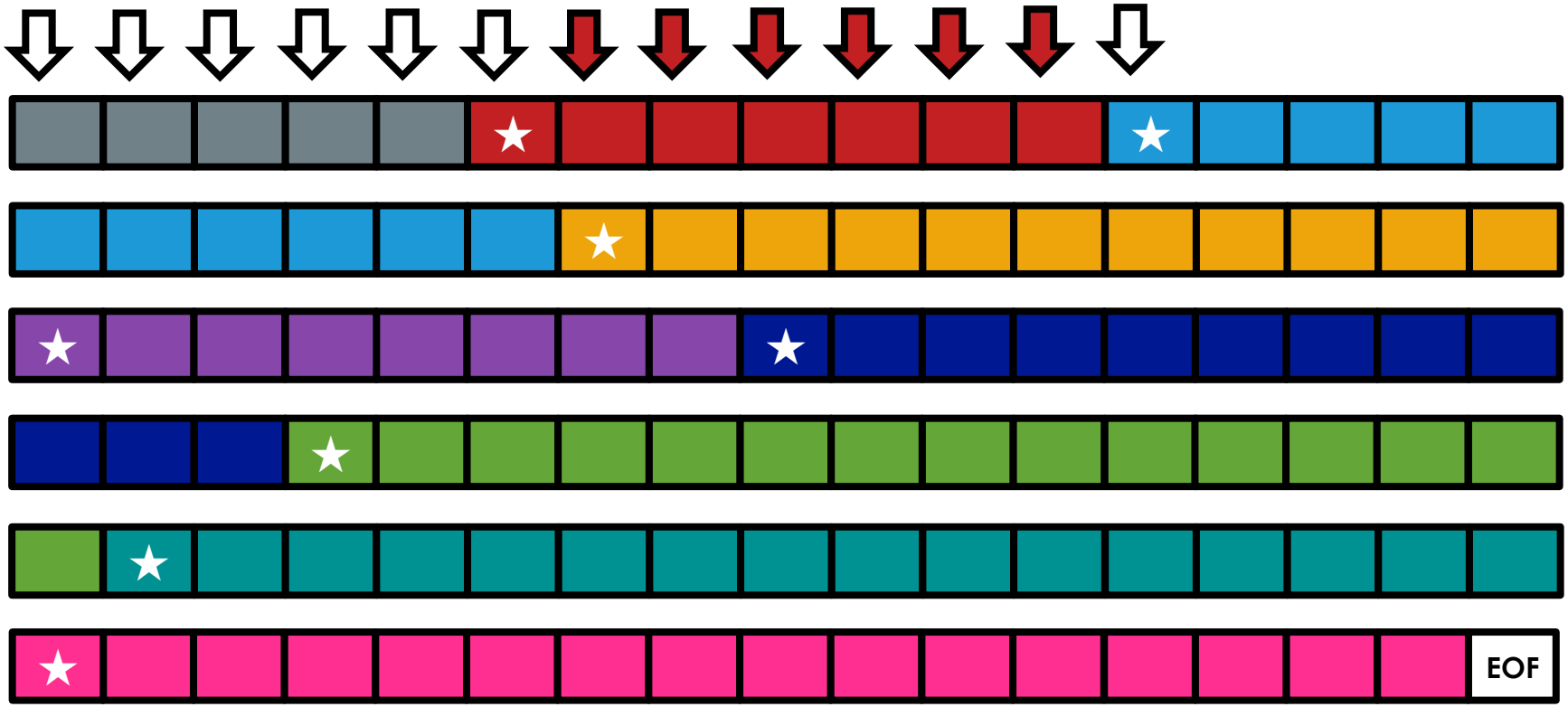


reading files

```
fread(data, size, number, inptr);
```

- `data`: pointer to a struct that will contain the bytes you're reading
- `size`: size of each element to read
  - `sizeof`
- `number`: number of elements to read
- `inptr`: `FILE *` to read from

memory card



reading files

- `fread` returns number of elements successfully read

```
fread(buffer, 1, 512, raw_file);
```

vs.

```
fread(buffer, 512, 1, raw_file);
```

JPEG?

- each JPEG starts with a distinct header:
  - first 3 bytes: `0xff 0xd8 0xff`
  - last byte: `0xe0, 0xe1, 0xe2, ... 0xef`

… this could get messy

JPEG?

```
if (buffer[0] == 0xff &&
    buffer[1] == 0xd8 &&
    buffer[2] == 0xff &&
    (buffer[3] & 0xf0 == 0xe0)
```

TODO

- ☑ open memory card file
- ☑ find beginning of JPEG
- ☐ open a new JPEG
- ☐ write 512 bytes until new JPEG is found
- ☐ detect of file

making a new JPEG

- filenames: `###.jpg`
- named in the order in which they are found, starting at 000.
  - (so keep track!)

making a new JPEG

- `sprintf(filename,"%03i.jpg", 2);`
  - `filename`: char array to store the resultant string
    - `002.jpg`
- `FILE *img = fopen(filename, "w");`

TODO

- ☑ open memory card file
- ☑ find beginning of JPEG
- ☑ open a new JPEG
- ☐ write 512 bytes until new JPEG is found
- ☐ detect end of file

writing files

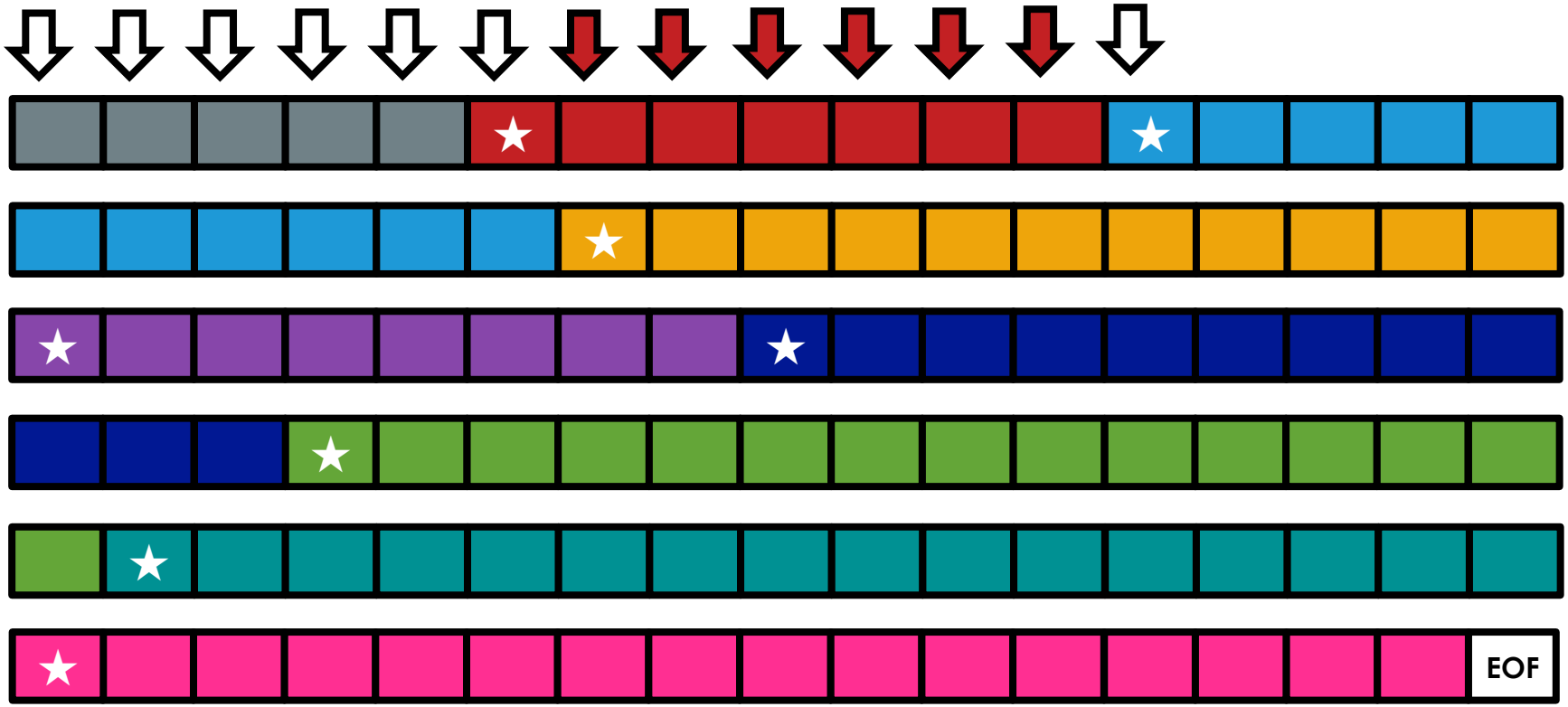
```
fwrite(data, size, number, outptr);
```

- `data`: pointer to the struct that contains the bytes you're reading from
- `size`
- `number`
- `outptr`: `FILE *` to write to

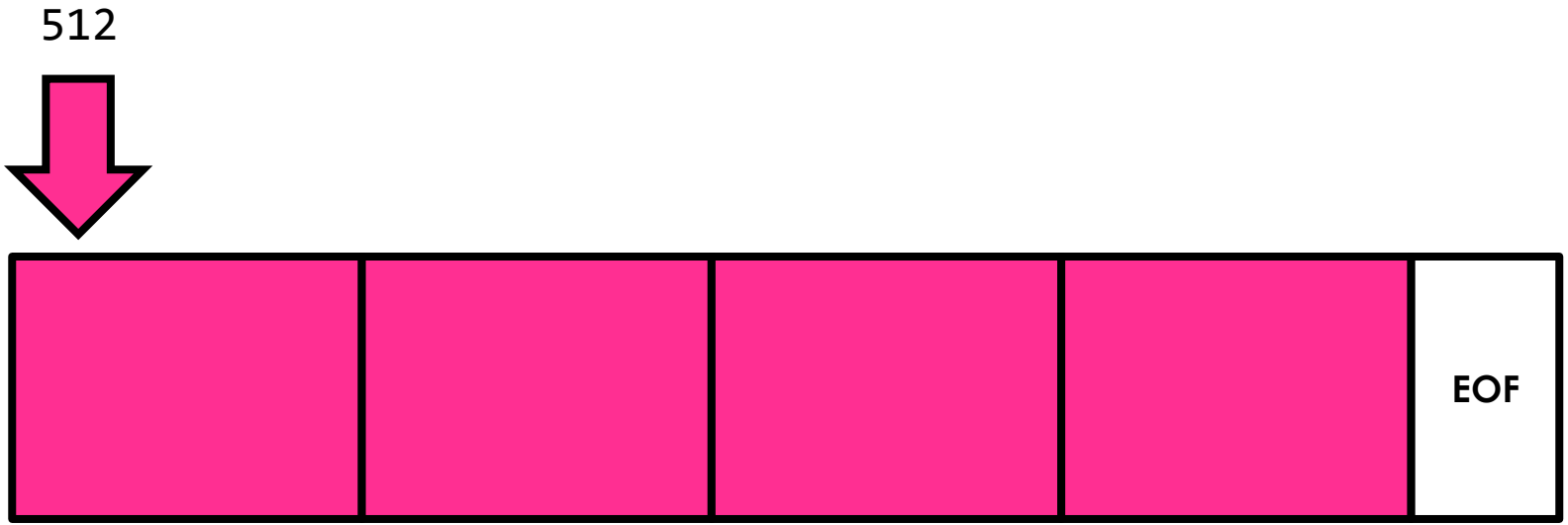
TODO

- ☑ open memory card file
- ☑ find beginning of JPEG
- ☑ open a new JPEG
- ☑ write 512 bytes until new JPEG is found
- ☐ detect end of file

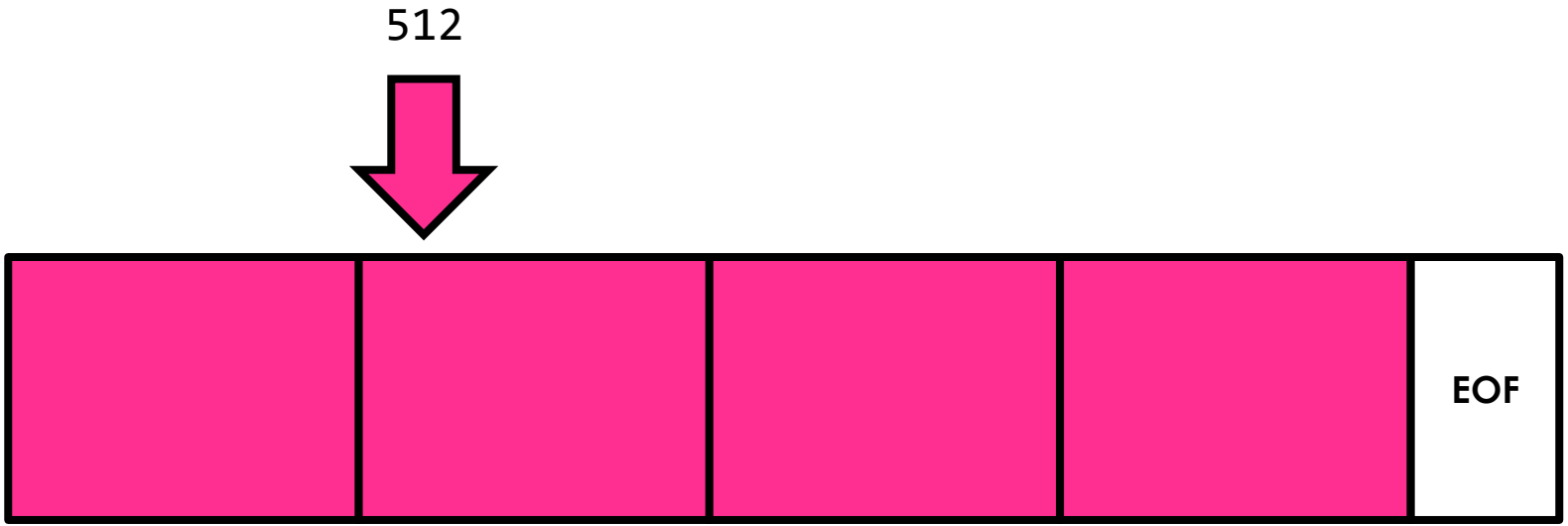
memory card



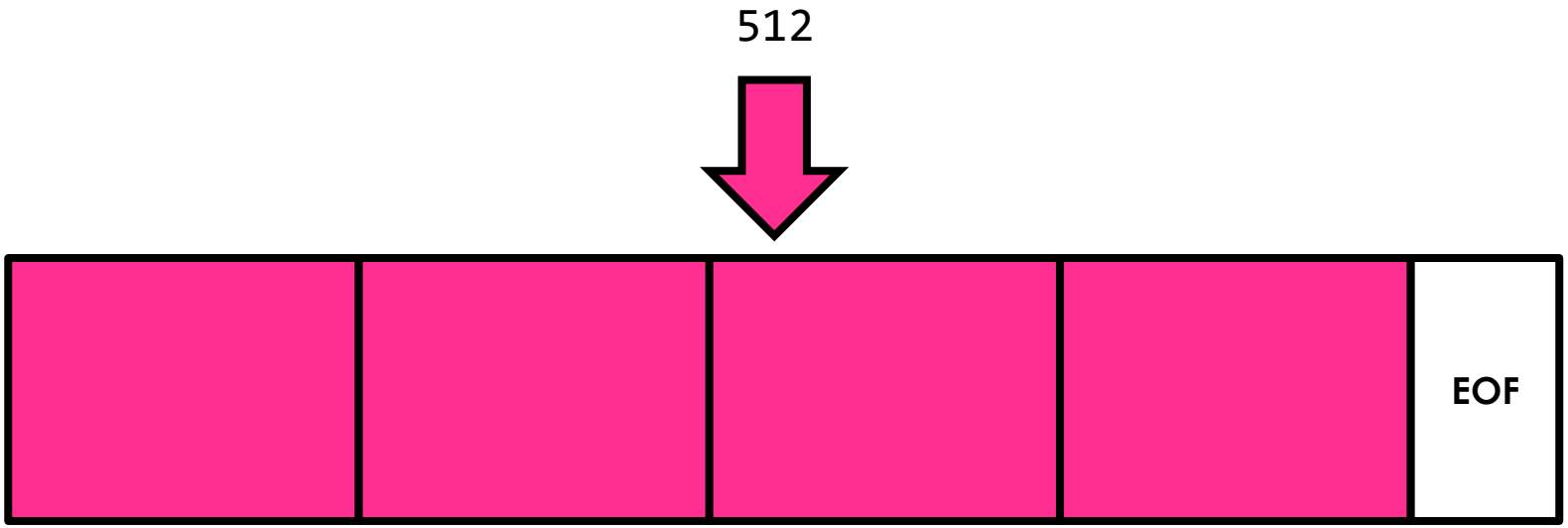
memory card



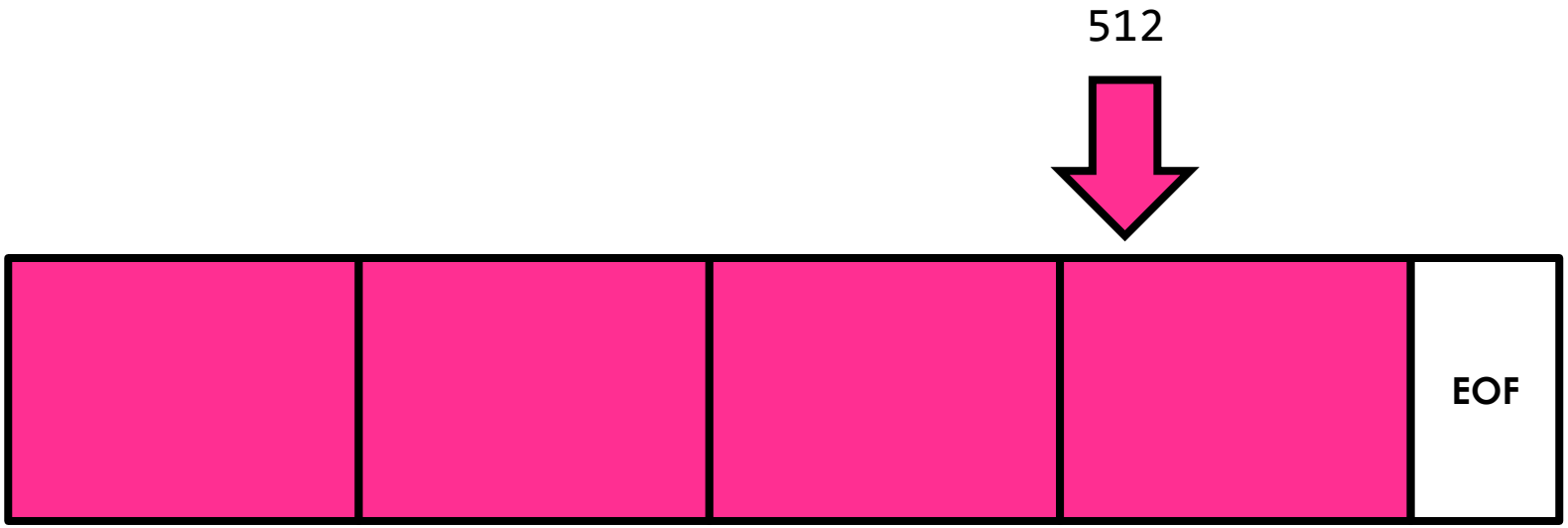
memory card



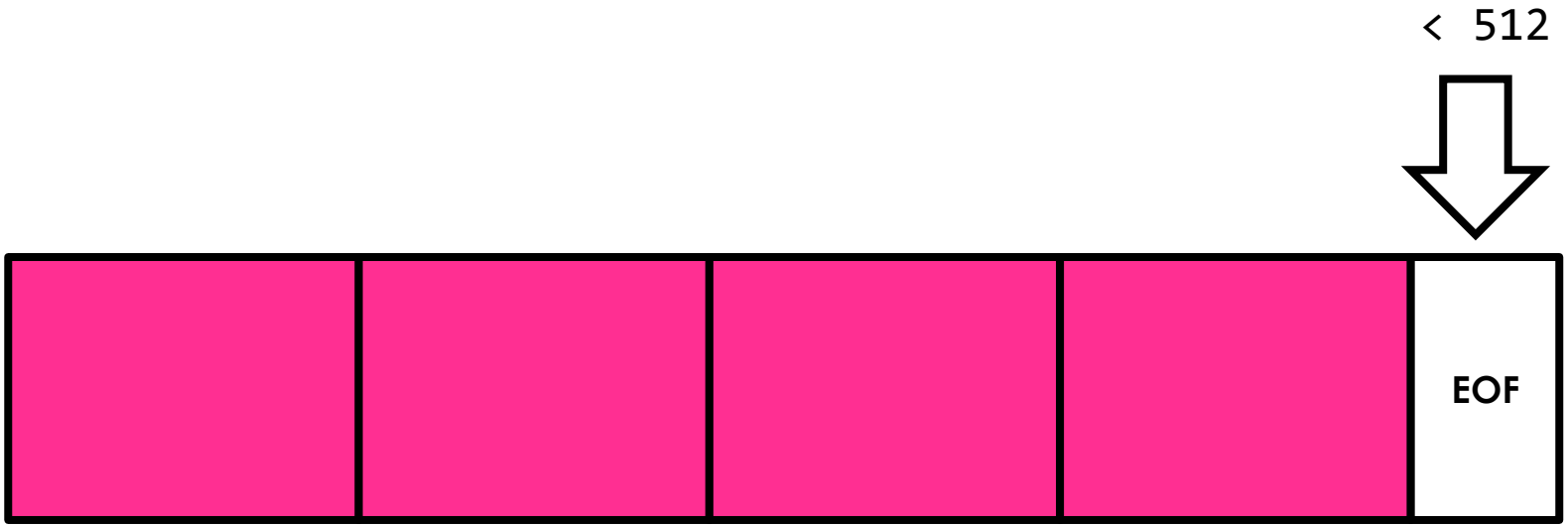
memory card



memory card



memory card



end of file?

- `fread` returns number of elements successfully read

```
fread(buffer, 1, 512, raw_file);
```

vs.

```
fread(buffer, 512, 1, raw_file);
```

reading files

```
fread(data, size, number, inptr);
```

- returns how many items of size `size` were read
  - (and ideally, it returns `number`)
- use it directly in a condition!

pseudocode

open card file

repeat until end of card

    read 512 bytes into a buffer

    start of a new JPEG?

        yes → …

        no → …

    already found a JPEG?

        no → …

        yes → …

close any remaining files

this was recover