# Questions

## What's `stdint.h`?

\**stdint.h is a header file in the C standard library to allow programmers to write more portable code by providing a set of typedefs that specify exact-width integer types, together with the defined minimum and maximum allowable values for each type, using macros .*\*

## What's the point of using `uint8\_t`, `uint32\_t`, `int32\_t`, and `uint16\_t` in a program?

\**The typedef name int N \_t shows a signed integer type with width N, no padding bits, and a two's-complement representation. Thus, int8\_t denotes a signed integer type with a width of exactly 8 bits.*\*

\**The typedef name uint N \_t designates an unsigned integer type with width N. Thus, uint16\_t denotes an unsigned integer type with a width of exactly 16 bits.*\*

## How many bytes is a `BYTE`, a `DWORD`, a `LONG`, and a `WORD`, respectively?

BYTE: 1 byte

DWORD: 4 byte

LONG: 8byte

WORD: 2byte

## What (in ASCII, decimal, or hexadecimal) must the first two bytes of any BMP file be? Leading bytes used to identify file formats (with high probability) are generally called "magic numbers."

The first 2 bytes of the BMP file format are the character “B” then the character “M” in ASCII encoding. Letters are represented by their ASCII value. ‘B’ is decimal 66 (hex 42). ‘M’ is decimal 77 (hex 4d). This is 0x42 and 0x4D respectively in Hexadecimal.

## What's the difference between `bfSize` and `biSize`?

'bfSize' means size of the bitmap file's size in bytes.

'biSize' means structures' required bytes.

\**The difference between the two is that bfSize is the whole bitmap file's size whereas biSize is the number of bytes needed just by the BITMAPINFOHEADER.*\*

## What does it mean if `biHeight` is negative?

If biHeight is negative, the bitmap is a top-down DIB and its origin is the upper-left corner.

Also, indicating a top-down DIB, biCompression must be either BI\_RGB or BI\_BITFIELDS. Top-down DIBs cannot be compressed.

## What field in `BITMAPINFOHEADER` specifies the BMP's color depth (i.e., bits per pixel)?

\**The count of bits-per-pixel. The biBitCount member of the BITMAPINFOHEADER structure determines the count of bits-per-pixel; these define each pixel and the maximum number of colors in the bitmap.*\*

## Why might `fopen` return `NULL` in lines 24 and 32 of `copy.c`?

\*Input file's pointer can not allocate memory to create the desired file if RAM is not available then it shows nowhere and nowhere means NULL.

Also, there may be permission error when trying to write to an open file, or you may not have access.\*

## Why is the third argument to `fread` always `1` in our code? (For example, see lines 40, 44, and 75.)

\**The fread() function reads 1 block of sizeof(BITMAPINFOHEADER), which is 40, and then it will be read from intptr written into*&*bi where it specifies for the pointer to the block of memory to. It will always read 1 in our code as it.*\*

## What value does line 63 of `copy.c` assign to `padding` if `bi.biWidth` is `3`?

int padding = (4 - ( 3 \* 3 ) ) %4 ) % 4 = 3;

int padding = 3;

## What does `fseek` do?

\**fseek function allows you rewind or fast-forward within a file.*\*

## What is `SEEK\_CUR`?

It means cursor of the seeked file. It denotes file pointer's current position.