Operation	Result
a	[01001110]
b	[11100001]
$\sim a$	[10110001]
$\sim b$	[00011110]
$a \,\&\, b$	[01000000]
$a \mid b$	[11101111]
$a \,\hat{}\, b$	[10101111]

This problem illustrates how Boolean algebra can be used to describe and reason about real-world systems. We can see that this color algebra is identical to the Boolean algebra over bit vectors of length 3.

A. Colors are complemented by complementing the values of R, G, and B. From this, we can see that white is the complement of black, yellow is the complement of blue, magenta is the complement of green, and cyan is the complement of red.

B. We perform Boolean operations based on a bit-vector representation of the colors. From this we get the following:

Blue (001)	`|`	Green (010)	=	Cyan (011)
Yellow (110)	`&`	Cyan (011)	=	Green (010)
Red (100)	`^`	Magenta (101)	=	Blue (001)

This problem highlights the relation between bit-level Boolean operations and logical operations in C. A common programming error is to use a bit-level operation when a logical one is intended, or vice versa.

Expression	Value	Expression	Value
`a & b`	`0x44`	`a && b`	`0x01`
`a | b`	`0x57`	`a || b`	`0x01`
`~a | ~b`	`0xBB`	`!a || !b`	`0x00`
`a & !b`	`0x00`	`a && ~b`	`0x01`