**Binary Numbers**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  |  |  |  |  |
| 16 | 8 | 4 | 2 | 1 |
| 0 | 1 | 1 | 0 | 0 |

= 12

**Binary Operations**

A = 0011 1100

B = 0000 1101

A & B : 1 if both numbers have are a 1, 0 else

A & B = 0000 1100

A | B : 1 if it exists in either, 0 else

A | B = 0011 1101

A ^ B: 1 if it is 1 in either, 0 else or set to 1 in both

A ^ B = 0011 0001

A ~ B: flips the bits

A ~ B = -0111 101

A << #: everything moved left by #

A << 2 = 1111 0000

A >> #: everything moved right by #

A >> 2 = 0000 1111

**Pointer Arrays**

int nums[2][3] = { {16, 18, 20}, { 25, 26, 27} }

\*(\*nums) == nums[0][0] = 16

\*(\*nums + 1) == nums[0][1] = 18

\*(\*nums + 2) == nums[0][2] = 20

\*(\*(nums + 1)) == nums[1][0] = 25

\*(\*(nums + 1) + 1) == nums[1][1] = 26

\*(\*(nums + 1) + 2) == nums[1][2] = 27