

check bit-wise operation

you will need
bit or |
bit and &
bit negation ~
and bit shift <<, >>

eg, if you want to set 2 most significant bits of a int16 without changing other bits, you can:

```
void test_bit()
{
    uint16 target=0;
    DUMP(target);

    // set 2 most significant bits

    target |= ((uint16)0x3 << 14);
    // 0x3=0000 0000 0000 0011b right shift by 14 bits becomes 1100 0000 0000 0000b
    // or 0xC000
    DUMP(target);

    // and set the 3 LSBs
    target |= 0x7;
    DUMP(target);

    // if you instead want to unset the 2 least significant bits
    // you can do something like:
    target &= ~0x3;
    // 0x3 is 0000 0000 0000 0011b
    // ~0x3 becomes 1111 1111 1111 1100b
    // bitwise-and this number with target result in the
    // 2 LSB being unset.
    DUMP(target)
}
```