

GET THE NTH BIT

GET THE NTH BIT

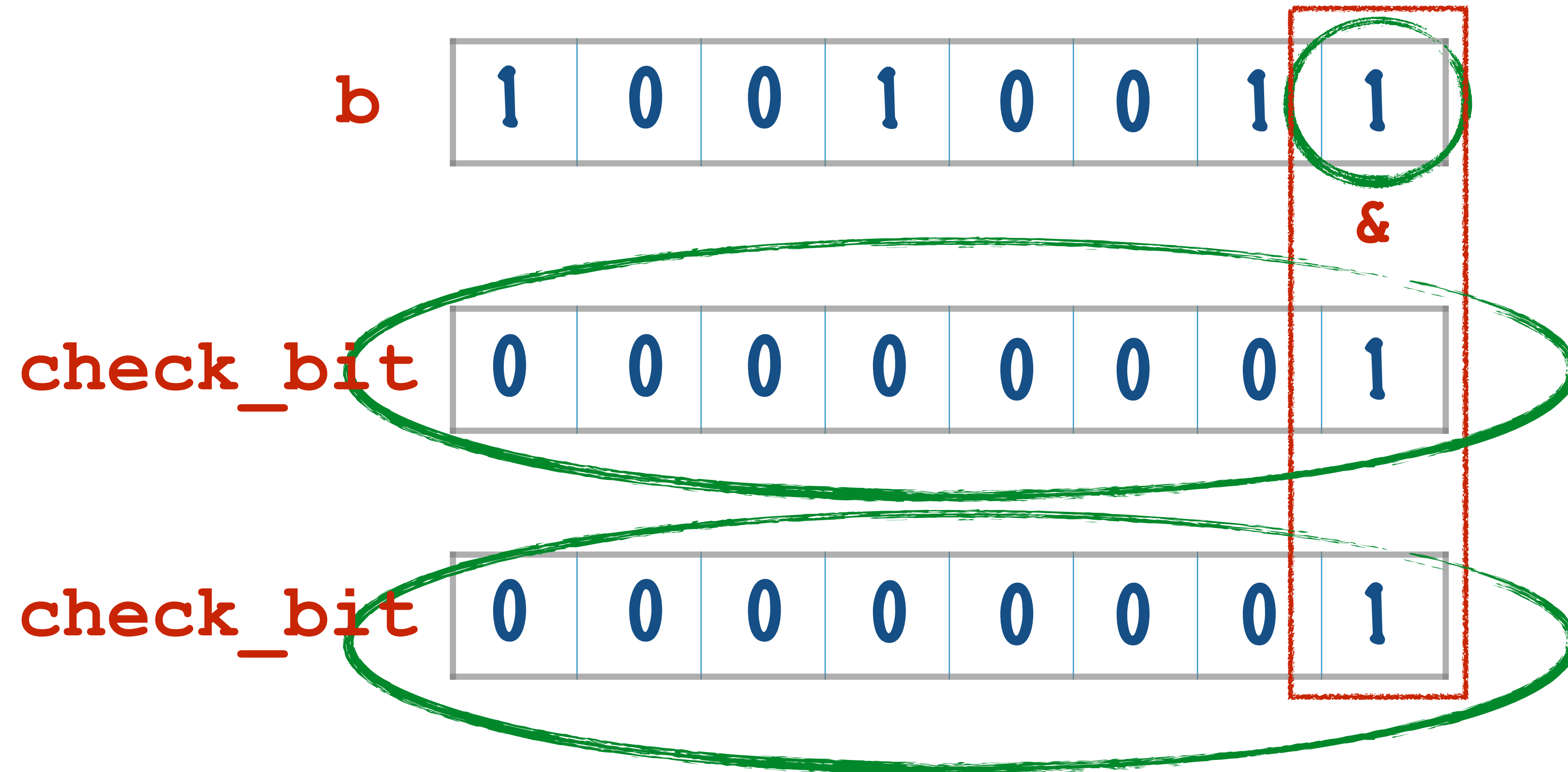
CHECK THE NTH BIT IN AN INTEGER

RETURN **1** IF IT IS 1 AND **0** IF IT IS 0

HOW DO YOU IDENTIFY A SPECIFIC
BIT IN A SERIES OF BITS?

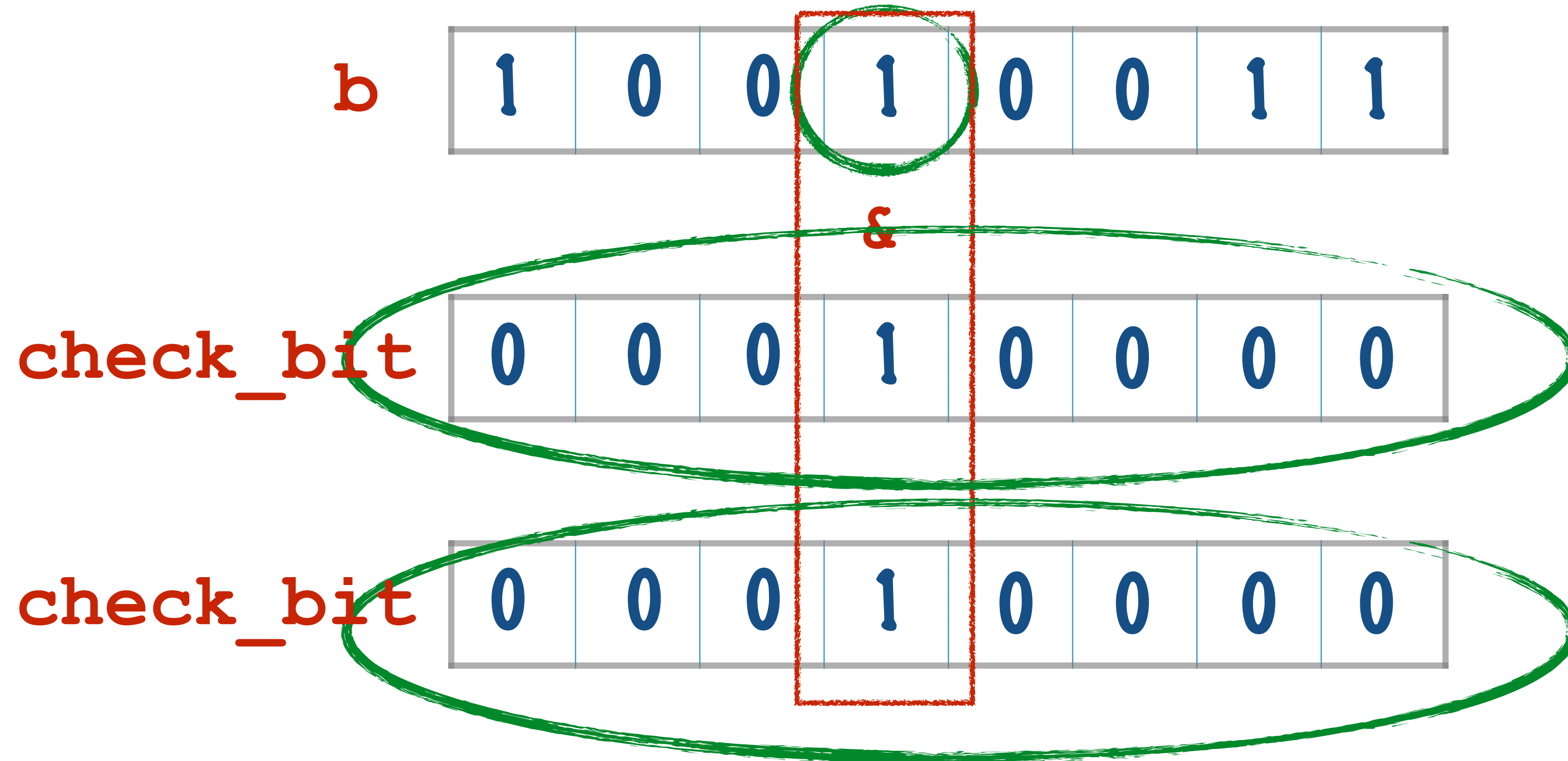
GET THE NTH BIT

CHECKING THE 0TH BIT - OR THE BIT
AT POSITION 0



GET THE NTH BIT

CHECKING THE 4TH BIT - OR THE BIT
AT POSITION 4 FROM THE RIGHT



GET THE NTH BIT

HOW DO GET A 1 BIT TO A SPECIFIC POSITION?

USE SHIFT LEFT!

1



1 << 2



1 << 4



GET THE NTH BIT

```
int get_nth_bit(int num, int n) {  
    int check_bit = 1 << n;  
  
    int and_bit = num & check_bit;  
    if (and_bit == check_bit) {  
        return 1;  
    }  
  
    return 0;  
}
```

MOVE THE CHECK BIT
TO THE NTH POSITION



BITWISE AND (&) WITH
THE NUMBER TO CHECK



IF THE RESULT IS THE SAME AS THE
ORIGINAL CHECK BIT - THEN THE BIT
AT THE NTH POSITION WAS 1



OTHERWISE IT IS 0



SET THE NTH BIT TO 1

SET THE NTH BIT TO 1

REMEMBER THE BIT MANIPULATION
TECHNIQUES

IF A BIT AT A POSITION IS OR'ED (|)
WITH BIT 1 AT THE SAME POSITION
THEN THAT PARTICULAR BIT'S VALUE
WILL BECOME 1

THIS IS TRUE WHATEVER THE
ORIGINAL VALUE OF THAT BIT

SET THE NTH BIT TO 1

SET THE 5TH BIT COUNTING FROM
THE RIGHT TO 1

num

1	0	0	1	0	0	1	1
---	---	---	---	---	---	---	---

set_bit

0	0	1	0	0	0	0	0
---	---	---	---	---	---	---	---

result

1	0	1	1	0	0	1	1
---	---	---	---	---	---	---	---

SET THE NTH BIT TO 1

```
int set_nth_bit_to_1(int num, int n) {  
    int set_bit = 1 << n;  
  
    int result = num | set_bit;  
  
    return result;  
}
```

MOVE THE SET BIT TO
THE NTH POSITION



BITWISE OR (|) WITH
THE NUMBER TO CHECK



RETURN THE RESULT

