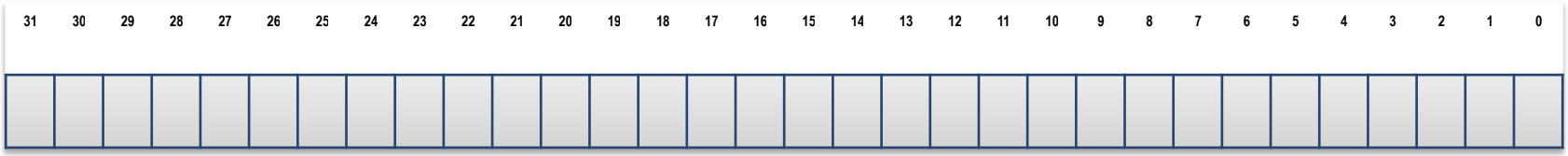
ECE 2534

Bit manipulation in C

Integer representation

- On our system, integers are (typically) represented using 32 bits

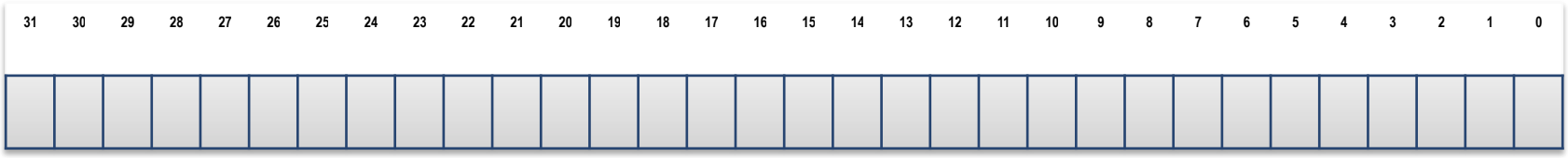


- For signed values, we (typically) assume 2's-complement representation

Integer representation

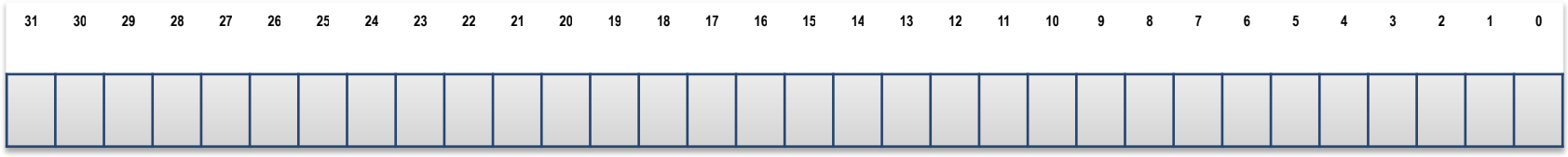
- Example:

```
int i = 17;
```



- Example:

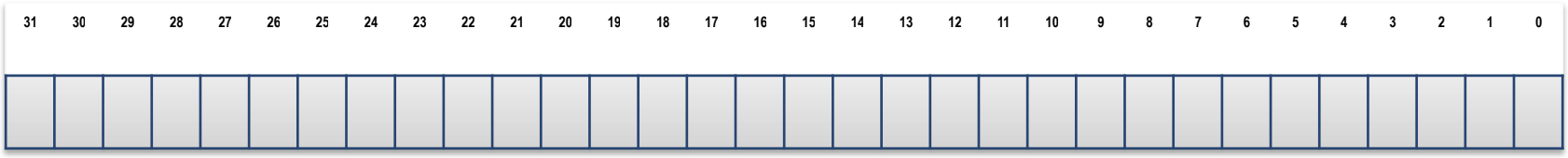
```
int j = -2;
```



Integer representation

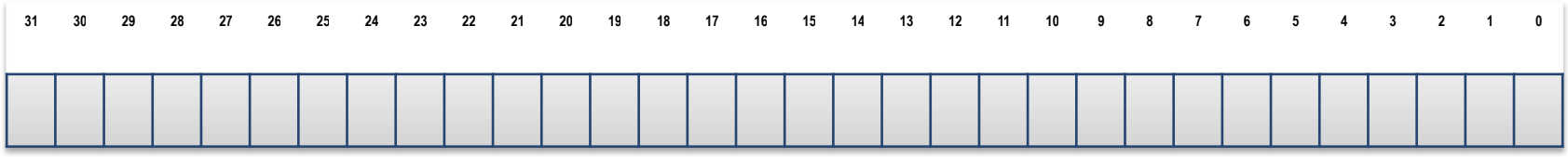
- Example:

```
int i = 0x23;
```



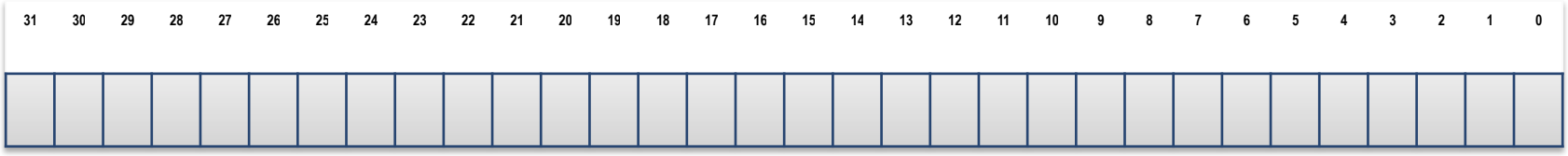
- Example:

```
int j = 0xf00d;
```



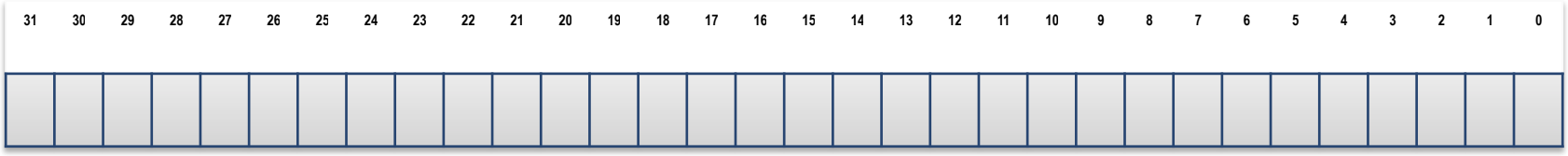
Bit manipulation

- Problem: Set bit 12 to 1, and clear all other bits to 0



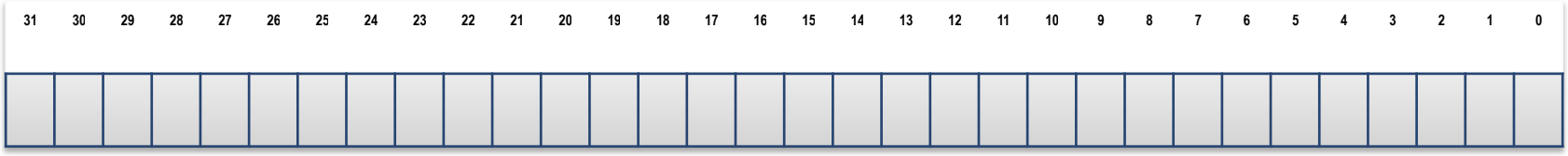
Bit manipulation

- Problem: Set bit 12 to 1, and *leave all other bits unchanged*



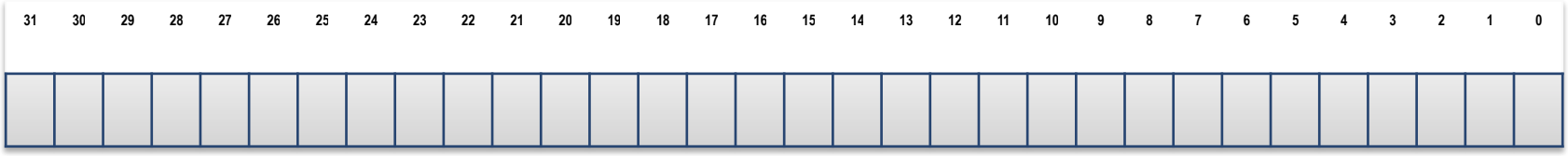
Bit manipulation

- Problem: Set bits 12 through 14 to 1, and *leave all other bits unchanged*



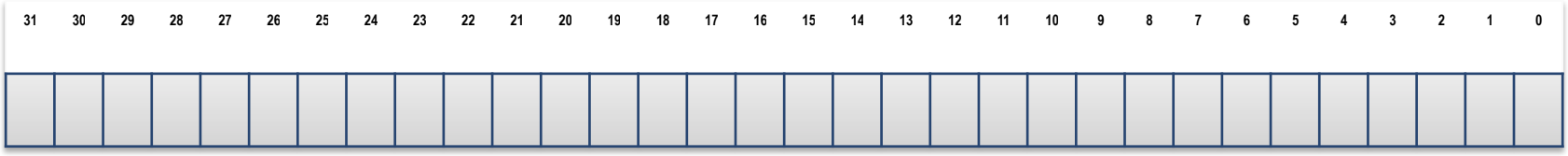
Bit manipulation

- Problem: Clear bit 12 to 0, and *leave all other bits unchanged*



Bit manipulation

- Problem: Complement bit 12, and *leave all other bits unchanged*



Conditional expressions

- What does this do?

```
if ((i == 2) && (j == 3))
   . . .
```

- What does this do?

```
if (n & 0x1000)
   . . .
```

Summary

- Bit manipulation will be needed through this course
- Examples:

```
int ioReg, n;
. . .

ioReg = ioReg | (1 << n);   // sets bit n
ioReg = ioReg &  ~(1 << n); // clears bit n
ioReg = ioReg ^ (1 << n);   // toggles bit n

if (ioReg & (1 << n))       // tests bit n
   . . .
```