COMP1511 17s2

— Lecture 3 —

Count On It

Andrew Bennett
<andrew.bennett@unsw.edu.au>

review: decisions and conditions
counting and number systems

While you wait...

Go to the course website, and answer the polls!

`webcms3.cse.unsw.edu.au/COMP1511/17s2`

Admin

Don't panic!

swapping tute-lab times

lecture recordings are on WebCMS 3

make sure you have **home computing** set up
VLAB

the style guide

Making Decisions, Redux

```
if (condition 1) {
    // Do stuff
} else if (condition 2) {
    // Do something else
} else if (condition 3) {
    // Do something completely different
} else {
    // In all other cases, do this.
}
```

Indentation


```
if (condition 1) {
// Do stuff
} else if (condition 2) {
// Do something else
} else if (condition 3) {
// Do something completely different
} else {
// In all other cases, do this.
}
```

Indentation


```
if (condition 1) {
    // Do stuff
} else if (condition 2) {
    // Do something else
} else if (condition 3) {
    // Do something completely different
} else {
    // In all other cases, do this.
}
```

Indentation


```
if (condition 1) {
if (condition 2) {
if (condition 3) {
// Do stuff
} else if (condition 4) {
// Do something else
}
} else if (condition 5) {
if (condition 6) {
// Do something completely different
}
} else {
// In all other cases, do this.
}
}
```

Indentation

```
if (condition 1) {
    if (condition 2) {
        if (condition 3) {
            // Do stuff
        } else if (condition 4) {
            // Do something else
        }
    } else if (condition 5) {
        if (condition 6) {
            // Do something completely different
        }
    } else {
        // In all other cases, do this.
    }
}
```

revisiting `driving2.c`

is “it works” enough?

number systems

you can count on it!

Number Systems

usually, we count in base 10,
but that's not always been the case

other common bases:
base 60, the Sumerians/Babylonians;
minutes in an hour

base 12, hours of the day

computers work in two-state logic: **on** and **off**,
which is easy with two voltage levels and logic gates;
so we often use bases that are powers of 2

base 2, or binary,
base 8, or octal,
base 16, or hexadecimal.

Decimal Representations

we can interpret the decimal number 4705 as:

$$4 \times 10^3 + 7 \times 10^2 + 0 \times 10^1 + 5 \times 10^0$$

the base or radix is 10;
in each place, the digits 0 through 9
each place has a value of 10^n
1000 is 10^3
100 is 10^2
10 is 10^1
1 is 10^0

we often write numbers with the radix as a subscript, like 4705_{10}

Binary Representations

in a similar way,
we can interpret a binary number 1011 as:

$$1 \times 2^3 + 0 \times 2^2 + 1 \times 2^1 + 1 \times 2^0$$

the base or radix is 2;
in each place, the digits 0 or 1
each place has a value of 10^n
8 is 2^3
4 is 2^2
2 is 2^1
1 is 2^0

and we would write 1011_2
$(= 11_{10})$

$$2^0 = 1$$
$$2^1 = 2$$
$$2^2 = 4$$
$$2^3 = 8$$
$$2^4 = 16$$
$$2^5 = 32$$
$$2^6 = 64$$
$$2^7 = 128$$
$$2^8 = 256$$
$$2^9 = 512$$
$$2^{10} = 1024$$

that’s exponential!

Binary to Decimal

$$
\begin{aligned}
&1011_2 \\
&= 1 \times 2^3 + 0 \times 2^2 + 1 \times 2^1 + 1 \times 2^0 \\
&= 1 \times 8 + 0 \times 4 + 1 \times 2 + 1 \times 1 \\
&= 8 + 2 + 1 \\
&= 11_{10}
\end{aligned}
$$

Decimal to Binary

$$221_{10}$$

$$= 128 + 101$$

$$= 128 + 64 + 37$$

$$= 128 + 64 + 32 + 5$$

$$= 128 + 64 + 32 + 4 + 1$$

$$= 2^7 + 2^6 + 2^5 + 2^2 + 2^0$$

$$= 11100101_2$$

Octal Representations

we can interpret the octal number 1511 as:

$$1 \times 8^3 + 5 \times 8^2 + 1 \times 8^1 + 1 \times 8^0$$

the base or radix is 8;
in each place, the digits 0 through 7
each place has a value of 8^n
512 is 8^3
64 is 8^2
8 is 8^1
1 is 8^0

and we would write $\texttt{1511}_8$
$(= 841_{10})$

Binary to Octal to Binary

3 binary digits ⇔ 1 octal digit

$$
\begin{aligned}
000_2 &= 0_8 \\
001_2 &= 1_8 \\
010_2 &= 2_8 \\
011_2 &= 3_8 \\
100_2 &= 4_8 \\
101_2 &= 5_8 \\
110_2 &= 6_8 \\
111_2 &= 7_8
\end{aligned}
$$

Hexadecimal Representations

we can interpret the hexadecimal number 3AF1 as:

$$3 \times 16^3 + 10 \times 16^2 + 15 \times 16^1 + 1 \times 16^0$$

the **base** or **radix** is 16;
in each place, the digits 0 through 9 and A through F
each place has a value of 16^n
4096 is 16^3
256 is 16^2
16 is 16^1
1 is 16^0

and we would write $\mathtt{3AF1}_{16}$
$(= 15089_{10})$

Binary to Hexadecimal to Binary

4 binary digits ⇔ 1 hexadecimal digit

$$
\begin{aligned}
0000_2 &= 0_{16}\\
0001_2 &= 1_{16}\\
0010_2 &= 2_{16}\\
0011_2 &= 3_{16}\\
0100_2 &= 4_{16}\\
0101_2 &= 5_{16}\\
0110_2 &= 6_{16}\\
0111_2 &= 7_{16}\\
1000_2 &= 8_{16}\\
1001_2 &= 9_{16}\\
1010_2 &= \mathtt{A}_{16}\\
1011_2 &= \mathtt{B}_{16}\\
1100_2 &= \mathtt{C}_{16}\\
1101_2 &= \mathtt{D}_{16}\\
1110_2 &= \mathtt{E}_{16}\\
1111_2 &= \mathtt{F}_{16}
\end{aligned}
$$

Try it!

Convert these numbers to binary:

53_{10}

$\texttt{5F3A}_{16}$

$\texttt{12D}_{16}$

3701_8

4232_8

Convert these numbers to octal:

$10\,101\,111\,011_2$

$\texttt{5F3A}_{16}$

Convert these numbers to hexadecimal:

$10\,101\,111\,011_2$

3701_8

Try it!

Convert these numbers to binary:

$$53_{10} = 110101_2$$
$$\mathtt{5F3A}_{16} = 0101\,1111\,0011\,1010_2$$
$$\mathtt{12D}_{16} = 0001\,0010\,1101_2$$
$$3701_8 = 11\,111\,000\,001_2$$
$$4232_8 = 100\,010\,011\,010_2$$

Convert these numbers to octal:

$$10\,101\,111\,011_2 = 2573_8$$
$$\mathtt{5F3A}_{16} = 57\,472_8$$

Convert these numbers to hexadecimal:

$$101\,0111\,1011_2 = \mathtt{57B}_{16}$$
$$3701_8 = \mathtt{7C1}_{16}$$