Tommy
Skadje

CS 255 HW2: Number Systems

1.) a.) Since we are using binary, we can store 2^n patterns, where n is the number of bits. We have $3 \times 8 = 24$ bits, so we can store $\boxed{2^{24} \text{ patterns per integer.}}$

b.) For 2701282:

2701282 / 2 = 1350641 remainder 0. (0th bit = 0)
1350641 / 2 = 675320 remainder 1. (1st bit = 1)
675320 / 2 = 337660 r = 0 (2nd bit = 0)
337660 / 2 = 168830 r = 0 (3rd bit = 0)
168830 / 2 = 84415 r = 0 (4th bit = 0)
84415 / 2 = 42207 r = 1 (5th bit = 1)
42207 / 2 = 21103 r = 1 (6th bit = 1)
21103 / 2 = 10551 r = 1 (7th bit = 1)
10551 / 2 = 5275 r = 1 (8th bit = 1)
5275 / 2 = 2637 r = 1 (9th bit = 1)
2637 / 2 = 1318 r = 1 (10th bit = 1)
1318 / 2 = 659 r = 0 (11th bit = a)
659 / 2 = 329 r = 1 (12th bit = 1)
329 / 2 = 164 r = 1 (13th bit = 1)
164 / 2 = 82 r = 0 (14th bit = 0)
82 / 2 = 41 r = 0 (15th bit = 0)
41 / 2 = 20 r = 1 (16th bit = 1)
20 / 2 = 10 r = 0 (17th bit = 0)
10 / 5 = 5 r = 0 (18th bit = 0)
5 / 2 = 2 r = 1 (19th bit = 1)
2 / 2 = 1 r = 0 (20th bit = 0)

1.) b.) continued

1/2 = 0   r = 1   (21st bit = 1)

The 22nd and 23rd bits are left as 0.
So 2701282 in computer X is:

0010 1001 0011 0111 1110 0010

For -2701282

Since we are using two's complement, flip all the bits in 2701282 and then add 1 to the result.

So -2701282 in computer X is:

1101 0110 1100 1000 0001 1110

∴

1.) c.) i.) First convert 4D 7A A5 to binary.
$4 = 2^2 = 0100$. $D = 13 = 2^3 + 2^2 + 2^0 = 1101$
$7 = 2^2 + 2^1 + 2^0 = 0111$ $A = 10 = 2^3 + 2^1 = 1010$
A = 1010 (Same as above) $5 = 2^2 + 2^0 = 0101$
So 4D 7A A5 in binary is:
0100 1101 0111 1010 1010 0101

The first bit is 0, so the number is positive.
We can just convert to decimal.

$2^{22} + 2^{19} + 2^{18} + 2^{16} + 2^{14} + 2^{13} + 2^{12} + 2^{11} + 2^9 + 2^7 + 2^5$
$+ 2^2 + 2^0 = \boxed{5077669}$

1.) c.) ii.) FD 60 78

Convert to binary.

$F = 15 = 2^3 + 2^2 + 2^1 + 2^0 = 1111$ $D = 13 = 2^3 + 2^2 + 2^0 = 1101$

$6 = 2^2 + 2^1 = 0110$ $0 = 0000$ $7 = 2^2 + 2^1 + 2^0 = 0111$

$8 = 2^3 = 1000$

So FD 60 78 in binary is:

1111 1101 0110 0000 0111 1000

The number starts with a 1, so it is negative.

To convert it to decimal, flip the bits and add 1.

0000 0010 1001 1111 1000 0111

↓

0000 0010 1001 1111 1000 1000

Now compute the absolute value.

$$2^{17} + 2^{15} + 2^{12} + 2^{11} + 2^{10} + 2^9 + 2^8 + 2^7 + 2^3$$

$= 171912$

Add the sign
-171912

1.) d.) We already computed the values in binary. Adding them: 0100 1101 0111 1010 1010 0101

+ 1111 1101 0110 0000 0111 1000

0000 0001 0100 1010 1101 1011 0001 1101

continued

1.) d.) 0000.0001 0100 1010 1101 1011 0001 1101

In hex, this is

01 4A DB 1D

Converting this to decimal:

$$1 \times 16^6 + 4 \times 16^5 + 10 \times 16^4 + 13 \times 16^3 + 11 \times 16^2 + 1 \times 16^1 + 13 \times 16^0$$

$$= 21682973$$

✗ 2.) a.)   1001101
+ 0111001
10000110

2.) b.)   101101
×   1001
101101
101101000
110010101

2.) c.) For 2a the result is 8 bits = 1 byte,
so a char is the minimum datatype that could
hold the value.

For 2b the result is 9 bits > 1 byte
so a short is the minimum datatype
that can store the result.

3.)

$$
\begin{array}{r}
4302 \\
\times\ 2432 \\
\hline
14104 \\
23411 \\
33213 \\
14104 \\
\hline
23234014
\end{array}
$$

$2 \times 3 = 6 = 1 \times 5^1 + 1 \times 5^0$
11

$9 = 1 \times 5^1 + 4 \times 5^0$
14

$13 = 2 \times 5^1 + 3 \times 5^0$
23

$8 = 1 \times 5^1 + 3 \times 5^0$
13

$12 = 2 \times 5^1 + 2 \times 5^0$
22

$10 = 2 \times 5^1 + 0 \times 5^0$
20

$14 = 2 \times 5^1 + 4 \times 5^0$
24

4.) 'C' in ASCII is 0x43 in hex.
'S' in ASCII is 0x53 in hex.

Converting them to binary we get
'C' = 0100 0011. 'S' = 0101 0011

Multiplying them:

$$
\begin{array}{r}
0100\ 0011 \\
\times\ 0101\ 0011 \\
\hline
0100\ 0011 \\
0100\ 0011 \\
0100\ 0011 \\
0100\ 0011 \\
0100\ 0011 \\
\hline
0101011011001
\end{array}
$$

4.) continued

0101011011001

↗

This is longer than 8 bits, so the first 'bits' will be cut off until there are only 8 bits left.

1011 1001 will be stored in x.

To convert to decimal, first flip the bits since this is a two's complement negative number.
0100 0110.
Then add 1.
0100 0111
Find the absolute value and add the sign.
$2^6 + 2^2 + 2^1 + 2^0 = 71 \longrightarrow -71$.
So -71 will be stored in x

need stop character?

5.)a.) 'H' = 0x48 'e' = 0x65 'l' = 0x6C
'p' = 0x70 '!' = 0x21 'NUL' = 0x00
Converting to binary: 'H' = 0100 1000 'e' = 0110 0101
'l' = 0110 1100 'p' = 0111 0000 '!' = 0010 0001
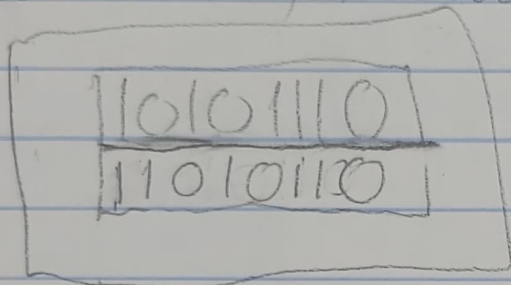'NUL' = 0000 0000 So in memory it would be.

0100 1000
0110 0101
0110 1100
0111 0000
0010 0001
0000 0000

5.) b.) AEDG in binary is:

1010 1110 1101 0110

Memory groups the numbers into 8-bit bytes
So in memory this would be:

```
| 1010 1110 |
| 1101 0110 |
```

⭐ 5.) c.) To convert -172 to two's complement,
first calculate 172 in binary.
172/2 = 86 rem. = 0 (0th bit = 0)
86/2 = 43 rem. = 0 (1st bit = 0)
43/2 = 21 rem. = 1 (2nd bit = 1)
21/2 = 10 rem. = 1 (3rd bit = 1)
10/2 = 5 rem. = 0 (4th bit = 0)
5/2 = 2 rem. = 1 (5th bit = 1)
2/2 = 1 rem. = 0 (6th bit = 0)
1/2 = 0 rem. = 1 (7th bit = 1)
A two's complement 8-bit number can only store
positive values up to $2^7 - 1 = 127$, so we need to
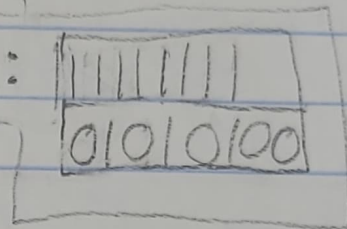pad another 8 bits. So 172 in two's complement is:
0000 0000 1010 1100

5.) C.) continued

0000 0000 1010 1100

Then flip all the bits and add 1.

1111 1111 0101 0011 → 1111 1111 0101 0100

In memory this would be: |1|1|1|1|1|1|1|1|
|0|1|0|1|0|1|0|0|

⭐ 5.) d.) 14.7

Convert 14 to binary first.

$14 - 2^3 = 6 - 2^2 = 2 - 2^1 = 0$

So 14 = 1110

Convert 0.7 to binary next.

$0.7 \times 2 = 1.4$ (1st bit 1)

$0.4 \times 2 = 0.8$ (2nd bit 0)

$0.8 \times 2 = 1.6$ (3rd bit 1)

$0.6 \times 2 = 1.2$ (4th bit 1)

$0.2 \times 2 = 0.4$ (5th bit 0)

This is the same as the 2nd bit, so it repeats

So $0.7 = .1\overline{0110}$. So $14.7 = 1110.1\overline{0110}$

In binary scientific notation this would be:

$1.1101\overline{0110} \times 2^3 = (1.1101\overline{0110}, 3)$

5.) d.) continued.
$$(1.11010110, 3)$$

For IEEE 754 floating point encoding, the first bit is the sign, which is 0 in this case since 14.7 is positive, the next 8 bits are the exponent, which are given the encoding of $(n-1)+128$. So
$$(3-1)+128 = 0000\ 0010$$
$$+\ \underline{1000\ 0000}$$
$$1000\ 0010$$

Finally, the last 23 bits are the mantissa, which is the numbers after the bicimal point in binary.

So 1101 0110 0110 0110 0110 011 ← truncated

So all together, 14.7 is

0 1000 0010 1101 0110 0110 0110 0110 011

In memory this would be:

0100 0001
0110 1011
0011 0011
0011 0011