Arithmetic Conversions

One of the basic skills a programmer needs is the ability to convert numbers between decimal, hexadecimal, and binary. This article will demonstrate the conversions between each of these bases.

Decimal to Hexadecimal

Here is an algorithm to convert a decimal integer to hexadecimal:

1) Let X be the decimal integer you wish to convert and let k = 1.

2) Divide X by 16, saving the quotient as Q, and the remainder (in hexadecimal) as R_k .

3) If Q is not zero, let X = Q, k = k + 1, and go back to step 2. Otherwise go to step 4.

4) Assume steps 1-3 were repeated n times. Arrange the remainders as a string of digits - $R_nR_{n-1}R_{n-2}\ldots R_3R_2R_1$.

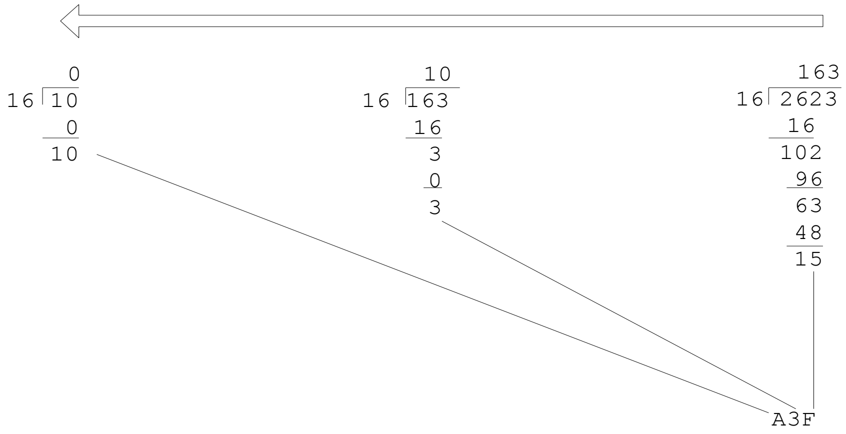
Here is an example where we number and describe each iteration of the algorithm:

1) Let X = 954 and let k = 1. Dividing by 16 we have Q = 59 and R_1 = A (10 = A in hexadecimal). Since Q is not zero, we let X = 59 and go back to start the second iteration.

2) Dividing X = 59 by 16, we have Q = 3 and R_2 = B since 11 = B in hexadecimal. Since Q is not zero, we let X = 3 and start the third iteration.

3) Dividing X = 3 by 16, we have Q = 0 and R3 = 3. Since Q = 0, the loop terminates and the algorithm is completed by arranging the remainders as “3BA”.

Here is another example using long division. In this example we convert 2623 in decimal to A3F in hexadecimal. The computations move from right to left.



Hexadecimal to Decimal

Consider the problem of converting the hexadecimal number A3B7 to decimal. Each hexadecimal digit in the representation represents a number of occurrences of a power of 16. We can change each hexadecimal digit to its decimal equivalent and multiply by the corresponding power of 16. Totaling the components produces the decimal representation of the original hexadecimal number.

$$\begin{array}{lcrcrcr}
A & = & 10 \text{ x } 16^3 & = & 10 \text{ x } 4096 & = & 40960 \\
3 & = & 3 \text{ x } 16^2 & = & 3 \text{ x } 256 & = & 768 \\
B & = & 11 \text{ x } 16^1 & = & 11 \text{ x } 16 & = & 176 \\
7 & = & 7 \text{ x } 16^0 & = & 7 \text{ x } 1 & = & 7 \\
 & & & & \text{Total} & = & 41911
\end{array}$$

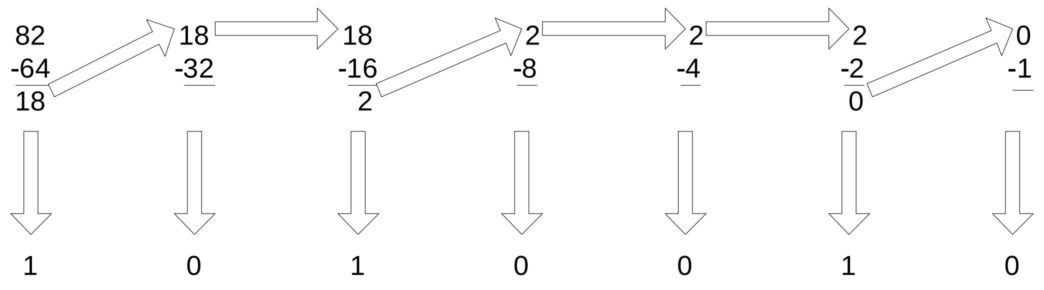
Here is another example in which 3FB9 is converted to decimal.

$$\begin{array}{lcrcrcr}
3 & = & 3 \text{ x } 16^3 & = & 3 \text{ x } 4096 & = & 12288 \\
F & = & 15 \text{ x } 16^2 & = & 15 \text{ x } 256 & = & 3840 \\
B & = & 11 \text{ x } 16^1 & = & 11 \text{ x } 16 & = & 176 \\
9 & = & 7 \text{ x } 16^0 & = & 9 \text{ x } 1 & = & 9 \\
 & & & & \text{Total} & = & 16313
\end{array}$$

Decimal to Binary

When a number is expressed in binary, each 1 in the expression represents a power of 2 (1, 2, 4, 8, …). Perhaps the faster way to convert from decimal to binary involves removing successive powers of two (starting with the largest and proceeding to the smallest) from the decimal number. Each time we are able to remove a power of 2, we record a 1 in the binary representation. Each time we are unable to remove a power of 2, we record a 0 in the binary representation. The operation proceeds from left to right. Consider the problem of converting the decimal integer 115 to binary. The largest power of 2 which can be removed from 115 is $2^6 = 64$. Removing 64 leaves 115 - 64 = 51. We record the removal of 64 as a 1 in the leftmost digit of the binary representation (1…). Next we remove 32 from 51. This leaves 51 - 32 = 19, and we record another 1 in the binary representation (11…). Continuing, we remove 16 from 19 leaving 3, and we record another 1 (111…). The next two powers of 2 are 8 and 4. These cannot be removed from 3 and so we record 0 in the next two positions (11100…). Next we remove 2 from 3 and then 1 from 1, recording 1 in the last two positions (1110011).

Here is an illustration where the decimal integer 82 is converted to the binary representation “1010010”. The computation proceeds from left to right with the successive removal of powers of 2. Notice that when a power of 2 is successfully removed it causes a “1” to occur in the binary representation. When we are unsuccessful, a “0” appears.



Binary to Decimal

In the representation of a binary integer, each 1 indicates the occurrence of the corresponding power of 2. To convert to decimal, we simply compute the total of the powers of 2 which correspond to a 1. Consider the following example in which we convert 1101110 to a decimal representation.

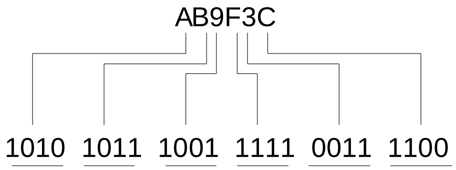
$$\begin{array}{lr}
1 = 1 \times 2^6 = & 64 \\
1 = 1 \times 2^5 = & 32 \\
0 = 0 \times 2^4 = & 0 \\
1 = 1 \times 2^3 = & 8 \\
1 = 1 \times 2^2 = & 4 \\
1 = 1 \times 2^1 = & 2 \\
0 = 0 \times 2^0 = & 0 \\
\text{Total} = & 110
\end{array}$$

Here is another example in which 100111 is converted to 39 in decimal.

$$\begin{array}{lr}
1 = 1 \times 2^5 = & 32 \\
0 = 0 \times 2^4 = & 0 \\
0 = 0 \times 2^3 = & 0 \\
1 = 1 \times 2^2 = & 4 \\
1 = 1 \times 2^1 = & 2 \\
1 = 1 \times 2^0 = & 1 \\
\text{Total} = & 39
\end{array}$$

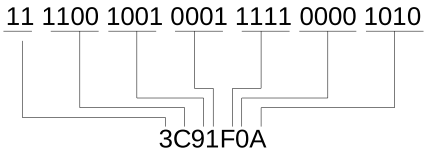
Hexadecimal to Binary

Base 16 and base 2 are “compatible” bases since 16 is a power of 2. This makes the conversion from hexadecimal to binary quite easy: Replace each hexadecimal digit with its four-bit binary representation. Here is an example in which the hexadecimal number AB9F3C is converted to 101010111001111100111100.



Binary to Hexadecimal

Converting from binary to hexadecimal involves replacing each consecutive block of four bits in the binary representation (moving from right to left within the representation) with its hexadecimal equivalent. You may need to add 0’s on the left of the binary representation in order that the number of 1’s and 0’s is a multiple of 4. Consider the following conversion.



11 1100 1001 0001 1111 0000 1010

3C91F0A

Each block of four bits beginning with 1010 is replaced with one hexadecimal digit. The leftmost 2 digits 11 are treated as 0011 and converted to “3”.