An IPv6 subnetting example:

You are issued a /48 block from your ISP. Subnet it so you have more than 500 networks.

Red numbers are the original 48 bits of the network issued by your ISP. The red bits do not change in this example.

Must borrow 9 more bits (2^9 > 500) to get more than 500 networks.

Blue numbers are newly borrowed network bits.

Green numbers are host bits. Counting the remaining 64 host bits, plus the 7 bits in hex values 15 & 16, there are 71 host bits in total.

71 host bits means each of the 500 networks that have been created will support 2^{71} hosts, or 2.36 x 10^{21} hosts.

Note that all IOS entries to test the IPv6 addresses in this spreadsheet were on a 3640 router running IOS 12.4.

```
    Hex    2    0    0    1   :  0    d    b    8   :  c    0    a    8  ::  0    0    0    0   : 0 0 0 0 : 0 0 0 0 : 0 0 0 0 : 0 0 0 0
  Binary  0010 0000 0000 0001 : 0000 1101 1011 1000 : 1100 0000 1010 1000 :: 0000 0000 0000 0000 :
```

After borrowing 9 bits, the red bits are unchanged, but now the subnetted blue bits are shown:

```
    Hex    2    0    0    1   :  0    d    b    8   :  c    0    a    8   :  0    0    0    0   :          ←   +64 more host bits   →
  Binary  0010 0000 0000 0001 : 0000 1101 1011 1000 : 1100 0000 1010 1000 : 0000 0000 0000 0000 :
 Decimal                                                                               0    1
  Binary                                                                              0000 0001
    Hex                                                                                0    1
 Decimal                                                                               0    2
  Binary                                                                              0000 0010
    Hex                                                                                0    2
 Decimal                                                                               0    3
  Binary                                                                              0000 0011
    Hex                                                                                0    3
 Decimal                                                                               0    4
  Binary                                                                              0000 0100
    Hex                                                                                0    4
 Decimal                                                                               0    5
  Binary                                                                              0000 0101
    Hex                                                                                0    5
 Decimal                                                                               0    6
  Binary                                                                              0000 0110
    Hex                                                                                0    6
 Decimal                                                                               0    7
  Binary                                                                              0000 0111
    Hex                                                                                0    7
 Decimal                                                                               0    8
  Binary                                                                              0000 1000
    Hex                                                                                0    8
 Decimal                                                                               0    9
  Binary                                                                              0000 1001
    Hex                                                                                0    9
 Decimal                                                                               0    10
  Binary                                                                              0000 1010
    Hex                                                                                0    A
 Decimal                                                                               0    11
  Binary                                                                              0000 1011
    Hex                                                                                0    B
```

Decimal	0	12
Binary	0000	1100
Hex	0	C
Decimal	0	13
Binary	0000	1101
Hex	0	D
Decimal	0	14
Binary	0000	1110
Hex	0	E
Decimal	0	15
Binary	0000	1111
Hex	0	F

Far right hextet reached max value, F, reset far right hextet to 0, increment next hextet by 1. Continue listing networks.

Hex	1	0
Binary	0001	0000
Decimal	1	1
Binary	0001	0001
Hex	1	1
Decimal	1	2
Binary	0001	0010
Hex	1	2
Decimal	1	3
Binary	0001	0011
Hex	1	3
Decimal	1	4
Binary	0001	0100
Hex	1	4
Decimal	1	5
Binary	0001	0101
Hex	1	5
Decimal	1	6
Binary	0001	0110
Hex	1	6
Decimal	1	7
Binary	0001	0111
Hex	1	7
Decimal	1	8
Binary	0001	1000
Hex	1	8
Decimal	1	9
Binary	0001	1001
Hex	1	9
Decimal	1	10
Binary	0001	1010
Hex	1	A
Decimal	1	11
Binary	0001	1011
Hex	1	B
Decimal	1	12

Binary	0001	1100
Hex	1	C
Decimal	1	13
Binary	0001	1101
Hex	1	D
Decimal	1	14
Binary	0001	1110
Hex	1	E
Decimal	1	15
Binary	0001	1111
Hex	1	F
•	•	•
•	•	•
•	•	•
	0111	0000
	7	0
•	•	•
•	•	•
	0111	1111
	7	F

< This address is valid to be assigned on an interface in IOS.

Once we reach this number, to increment again would change a blue bit, a network bit.
So the next network ID is listed on the next line:

```
     2    0    0    1  :  0    d    b    8  :  c    0    a    8  :  0    0    8    0
  0010 0000 0000 0001 : 0000 1101 1011 1000 : 1100 0000 1010 1000 : 0000 0000 1000 0000
```

< If I enter this address (with host bits all zero) on an interface, this error appears:
%2001:DB8:C0A8:80::/57 should not be configured on {int} a subnet router anycast
If I enter that same address and end the command with "anycast" no error appears.
eg:ipv6 add 2001:DB8:C0A8:80::/57 anycast
no error on the above command

The process repeats with green bits all zero until the green bits are all ones, then the blue bits would roll to 2, then 3, all the way up to F, then the blue bits would roll to

0000 0001 0000

and so on. This means the progression of network IDs would look like this:

0	0	0	0
0	0	8	0
0	1	0	0
0	1	8	0
0	2	0	0
0	2	8	0
•			
•			
•			
0	A	0	0
0	A	8	0
0	B	0	0
0	B	8	0
•			
•			
•			
0	F	0	0
0	F	8	0

If I enter ipv6 address 2001:DB8:C0A8:7F:FFFF:FFFF:FFFF:FFFF/57
I see this error:
{address} should not be configured on {int}, a reserved anycast

In both error cases above, the entered address was accepted on the interface in spite of the error.

```
                                                                             1    0    0    0
                                                                             1    0    8    0
                                                                             2    0    0    0
                                                                             2    0    8    0
                                                                             •
                                                                             •
                                                                             •
                                                                             F    0    0    0
                                                                             F    0    8    0
                                                                             F    1    0    0
                                                                             F    1    8    0
                                                                             •
                                                                             •
                                                                             •
                                                                             F    A    0    0
                                                                             F    A    8    0
                                                                             •
                                                                             •
                                                                             •
                                                                             F    F    0    0
                                                                             F    F    8    0
                                                    Once you reach FF80, all the
                                                    blue bits are now ones, which
                                                    means you have reached your
                                                    last subnet.  Note the blue
                                                    bits values:
           2    0    0    1  :   0    d    b    8  :   c    0    a    8  :   F    F    8    0    < Attempting to assign this address (with
         0010 0000 0000 0001 : 0000 1101 1011 1000 : 1100 0000 1010 1000 : 1111 1111 1000 0000   hosts all zero) results in error:
                                        Now allow the green bits to                              ...should not be configured on {int},
                                        roll through all their values                            a subnet router anycast
                                        in succession in the last subnet.
                                        The next value after FF80 is:        F    F    8    1    < This is not a subnet ID, it is one of the
                                                                           1111 1111 1000 0001   available IPv6 addresses.
                                                                             F    F    F    F
                                        Until all green bits set to one:   1111 1111 1111 1111   < This is not a subnet ID, it is one of the
                                        If you tried to continue, you                            available IPv6 addresses.
                                        would have to change a red bit,
                                        which is not allowed, you don't
                                        own those bits, the ISP does.
```

There are plenty of sites that allow you to input information and see the resulting subnet IDs.
I wanted to provide an example where you could actually see what happens to the individual bits.
Hit me with any comments,
Mark Jacob