Value	Range	128	64	32	16	8	4	2	1
Bits	Bits	1	1	1	1	1	1	1	1
Value + Bits To Left	Subnet Mask	128	192	224	240	248	252	254	255
2^N-2	Subnets	0	2	6	14	30	62	126	254

Bits	2 ^ N = x (Theoretical) - 2 = x (Useable)
1	2 ^ 1 = 2 – 2 = 0
2	2 ^ 2 = 4 – 2 = 2
3	2 ^ 3 = 8 – 2 = 6
4	2 ^ 4 = 16 – 2 = 14
5	2 ^ 5 = 32 – 2 = 30
6	2 ^ 6 = 64 – 2 = 62
7	2 ^ 7 = 128 – 2 = 126
8	2 ^ 8 = 256 – 2 = 254
9	2 ^ 9 = 512 – 2 = 510
10	2 ^ 10 = 1,024 – 2 = 1,022
11	2 ^ 11 = 2,048 – 2 = 2,046
12	2 ^ 12 = 4,096 – 2 = 4,094
13	2 ^ 13 = 8,192 – 2 = 8,190
14	2 ^ 14 = 16,384 – 2 = 16,382
15	2 ^ 15 = 32,768 – 2 = 32,766
16	2 ^ 16 = 65,536 – 2 = 65,534
17	2 ^ 17 = 131,072 – 2 = 131,070
18	2 ^ 18 = 262,144 – 2 = 262,142
19	2 ^ 19 = 524,288 – 2 = 524,286
20	2 ^ 20 = 1,048,576 – 2 = 1,048,574
21	2 ^ 21 = 2,097,152 – 2 = 2,097,150
22	2 ^ 22 = 4,194,304 – 2 = 4,194,302
23	2 ^ 23 = 8,388,608 – 2 = 8,388,606
24	2 ^ 24 = 16,777,216 – 2 = 16,777,214

Subnet Classes		
Class	High Order Bits	First Octet Range
A	0	1-126
B	10	128-191
C	110	192-223

Subnet Expansion						
Bit Pattern	Masked Bits	Provided Subnets	Subnet Mask	Class C Hosts/Subnets	Class B Hosts/Subnets	Class A Hosts/Subnets
11000000	2	2	.192	62	16,382	4,194,302
11100000	3	6	.224	30	8,190	2,097,150
11110000	4	14	.240	14	4,094	1,048,574
11111000	5	30	.248	6	2,046	524,286
11111100	6	62	.252	2	1,022	262,142
11111110	7	126	.254	0	510	131,070
11111111	8	254	.255	0	254	65,534

Subnetting Fundamentals:

1. **Subnetting creates smaller networks out of large networks** by borrowing host bits to create the subnet field, which is made up of the borrowed bits

2. **Subnetting increases network efficiency and security** by decreasing the size of broadcast domains

3. **Routers are used to create subnets** because OSI Layer 3 addressing (IP addresses) is used for subnetting

4. Students should be familiar with **network math and binary numbers** to do subnetting, and be able to **convert decimal numbers to binary numbers to at least 16 bits (two bytes)**

5. The formula $2^n-2=x$ will calculate the **number of subnets created** when n equals the number of host bits borrowed to create the subnet field

6. The formula $2^n-2=x$ will calculate the **number of hosts per subnet** created when n equals the number of remaining host bits after creating the subnet field

7. **Class A, Class B, and Class C networks can be subnetted**; Class D networks do not use a network ID, so the do not need to be subnetted; Class E networks are experimental

8. **Two host bits must remain after borrowing host bits to create subnets**, so the maximum number of host bits that can be borrowed in a Class A network is 22, in a Class B network is 14, and in a Class C network is 6

9. The **incremental value of subnets** is the decimal value of the last host bit borrowed to create the subnet field

10. When **expanding the subnet address ranges**, the first and last networks created are not usable; the host portion of the subnet address of the first subnet (normally listed as the 0 subnet) is all binary zeros and not usable; the host portion of the broadcast address on the last subnet is all binary ones, which makes that network unusable

11. When **expanding the subnet address ranges**, the first address will be the subnet address and the last address will be the broadcast address for that subnet; the useable host addresses will be the second address through the next to last address

12. **Incrementing subnets always starts in the octet of the last borrowed bit**; on a Class C network it is always in the fourth octet, on a Class B network incrementing can be in the third or fourth octet depending on which octet the last borrowed bit was in; on a Class A network incrementing can be in the second, third, or fourth octet depending on which octet the last borrowed but was in

Seven Steps to Subnetting

Excerpt from MCSE Guide to Microsoft® Windows 2000® Networking Certification Edition, *written by Kelly Caudle, Walter J. Glenn, and James Michael Stewart; published by Course Technology*

Creating Class C Subnetting Scheme

Basic subnetting is very easy when performed in seven steps. This example uses the Class C address 211.212.10.0. Using the seven steps provided here, you can create a subnetting scheme that allows you to use this address on your network.

Step 1: Determining Number of Subnets Needed

Determining the number of subnets you need is the very first step in subnetting. The number really depends upon your particular network. In Figure 2-3, the network consists of three routers connected via serial links. Each router also has a single Ethernet network attached.
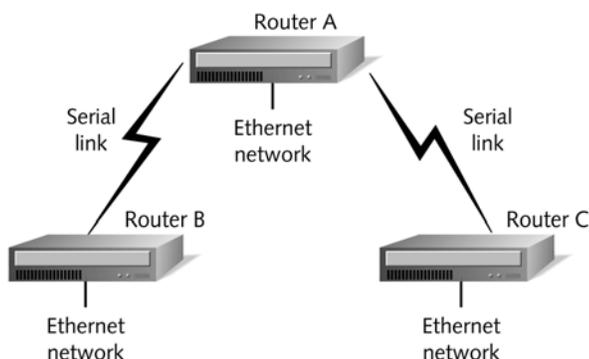
Figure 2-3   Example of network with three routers

Each shared serial link requires one subnet. Therefore, you need two subnets for the serial links between Router A and Routers B and C. You must also have one subnet per Ethernet interface on each router. Since you have three Ethernet networks, you need three subnets. Using this very simple counting method, you find that you need a total of five subnets. Unfortunately, you have been assigned a Class C address. The network address 211.212.10.0 allows for a single network of 254 hosts. You must borrow host ID bits to make this address work for you.

Step 2: Determining Number of Bits You Can Borrow

In Step 2, you must determine the number of bits that you can borrow. This number changes depending on the type of network address you start with. For Class A addresses, you have 24 host ID bits, but you can only borrow up to 22. For Class B addresses, you have 16 host ID bits, but you must have a minimum of two host bits; therefore, you can borrow 14 bits. Your Class C address (211.212.10.0) has eight total host ID bits, but you can only borrow a maximum of six. The easiest way to determine the number of bits you can borrow is to write the number of octets that contain host ID bits in binary. In the Class C example network 211.212.10.0, you have the following bits to "play" with:

        00000000

Step 3: Determining Number of Bits You Must Borrow to Get Needed Number of Subnets

After you determine the number of subnets you need and the number of bits you can borrow, you must calculate the number of host ID bits you must borrow to get the needed number of subnets. The formula for determining the number if bits you must borrow is 2^n-2= # of subnets. The *n* represents the number of bits you borrow. In other words, raise two to the power of the number of bits you borrow and subtract two from that number. The result is the number of useable subnets created when you borrow that number of bits. For the example network, you need five subnets. If you borrow three bits, the formula's result is six usable subnets: $2^3 = 8$-2 = 6.

Step 4: Turning On Borrowed Bits and Determining Decimal Value

In Step 4, using the bits you determined were available in Step 2, you turn on (set to 1) the number of bits determined you must borrow in Step 3. You must always begin with the high-order bits (the bits starting on the left of a binary number). Using the number of bits you can work with and the number of bits you must borrow (from Step 3), your result is the following: 11100000. In other words, from the eight total bits from Step 2 (six of which you could borrow), you borrow three host ID bits. In Step 4, you also need to determine the decimal value of the octets from which you borrow host ID bits. In this example, 11100000 equals 224. (128 + 64 + 32 = 224)

Step 5: Determining New Subnet Mask

Step 5 calculates the new subnet mask after you borrow the host ID bits in Step 4. You must add the decimal value from Step 4 to the default subnet mask for the class of address you are subnetting. The example is a Class C address, so the default mask is 255.255.255.0. The new mask after borrowing three bits becomes 255.255.255.224.

Step 6: Finding Host/Subnet Variable

In Step 6, you must find the lowest of the high-order bits (bits starting from the left) turned "on." Step 6 takes you all the way back to earlier in the chapter to the values found in each bit position within the octet. Our example defines the octets from which we borrow as 11100000. The highest order bit turned on represents 2^5, which equals 32. Since 2^5 is the last high-order bit turned on, the Host/Subnet variable you use in Step 7 is 32.

Step 7: Determining Range of Addresses

The final step allows you to take the Host/Subnet variable from Step 6 (32) and create your subnet ranges. Using the Class C network above, the range of subnets when you borrow three bits are:

~~211.212.10.0~~	~~to~~	~~211.212.10.31~~
211.212.10.32	to	211.212.10.63
211.212.10.64	to	211.212.10.95
211.212.10.96	to	211.212.10.127
211.212.10.128	to	211.212.10.159
211.212.10.160	to	211.212.10.191
211.212.10.192	to	211.212.10.223
~~211.212.10.224~~	~~to~~	~~211.212.10.255~~

IP addresses cannot be all ones or all zeros; therefore, in most cases the first range of addresses and the last range of addresses are unusable. (In some special circumstances, you can use the first range of addresses, or subnet 0. Only certain manufacturers' equipment, such as Cisco Systems, fully supports the use of subnet zero.) In each subnet, the first IP address is unusable because it represents the subnet ID. The final address is also unusable because it is the broadcast address for the subnet. Due to these two restrictions, in subnet one, 211.212.10.33 is the first useable host ID and 211.212.10.62 is the last useable host ID.

Tailoring a Class B Address

This example takes a Class B address and tries to fit it within the needs of a network containing 1000 subnets. You are assigned the Class B address 131.107.0.0. Using the following seven steps, you are going to subnet the Class B address to meet your needs.

Step 1: Determining Number of Subnets Needed

Examine your network and determine your needs based on current network configuration and future growth (in this case, 1000 subnets).

Step 2: Determining Number of Bits You Can Borrow

With this Class B network address, you have 16 total bits to work with. You can only borrow up to 14 of these. On your sheet of paper, you should write the number of bits you have in the host ID portion of the address:

00000000.00000000

Step 3: Determining Number of Bits You Must Borrow to Get Number of Subsets Needed

Using the formula $2^n-2=$ # of usable subnets, you can easily see that you need to borrow 10 bits. When you plug in 10 borrowed bits, you get the following result:

$2^{10} = 1024 - 2 = 1022$ useable subnets

Step 4: Turning on Borrowed Bits and Determining Decimal Value

If you turn on 10 bits, you get the following:

11111111.11000000

The decimal values for the octets are 255.192.

Step 5: Determining New Subnet Mask

Your example is a Class B address. In Class B addresses, the default subnet mask is 255.255.0.0. To get your new mask, you add the default mask to the decimal values found in Step 4. The new mask becomes:

255.255.255.192

Step 6: Finding Host/Subnet Variable

In the next-to-last step, you must find the value of the lowest high-order bit turned on in each octet, from which you borrowed host bits. Since this example is a Class B network and you must borrow a great number of bits to get the proper number of subnets, the borrowing crosses an octet boundary. As a result, you have two Host/Subnet variables. In this example, the variable in the third octet is 1, and the variable for the fourth octet is 64. You get these values by looking at the binary numbers in Step 4. The third octet has the final bit position, or the 2^0 bit position, turned on. Since $2^0 = 1$, your variable is 1 in the third octet. In the fourth octet, the second high-order bit or 2^6 is turned on. The variable in this octet is 64.

Step 7: Determining Range of Addresses

Figuring the range of addresses for Class B networks is much harder than for Class C. This is especially true in cases like this scenario in which you must borrow a large number of bits. Using 1 as the variable in the third octet and 64 as the variable in the fourth octet, the range of the first 9 subnets world be:

131.107.0.0	to	131.107.0.63
131.107.0.64	to	131.107.0.127
131.107.0.128	to	131.107.0.191
131.107.0.192	to	131.107.0.255
131.107.1.0	to	131.107.1.63
131.107.1.64.	to	131.107.1.127
131.107.1.128	to	131.107.1.191
131.107.1.192	to	131.107.1.255
131.107.2.0	to	131.107.2.63

Easy Subnetting
Adapted by Beau Sanders from an article originally published on the Internet by Greg Stanley, MCP

Let's begin by creating the following chart based on the formula **2^N -2 = X**:

First, write down a binary 1 for every bit in an octet:

1	1	1	1	1	1	1	1

Next, ABOVE each bit, write its **value**:

128	64	32	16	8	4	2	1
1	1	1	1	1	1	1	1

Now, BELOW each bit, write the **value of that bit plus the other bits to its left**:

128	64	32	16	8	4	2	1
1	1	1	1	1	1	1	1
128	192	224	240	248	252	254	255

Finally, BELOW all that, write the **value of 2^N-2**, starting from the left. This will give you the **number of subnets**:

128	64	32	16	8	4	2	1
1	1	1	1	1	1	1	1
128	192	224	240	248	252	254	255
0	2	6	14	30	62	126	254

Now lets see how this all works. In our first example we are given the Network ID: **192.62.123.0**

What subnet mask will you need for 10 subnets?

A quick look at the chart indicates that 224 will give you 6 subnets, but 240 yields 14 subnets. We need 10 subnets which is less than 14 and more than 6. The answer: **255.255.255.240**

All bits to the right of 240 (including other octets if Class A or B) make up the **Host ID**. The 240 and everything to the left of that compose the **Network ID**.

Now let's look at the **ranges of subnets**, also called **increments of subnets**. Look above 240 and you see **16**. This is the **first subnet address**, as well as the **range incrementing value**. The next subnet address will be 16 + 16, or 32. The next will be 32 + 16, or 48, and so on. The first address in each range is the address of the subnet itself (the Subnet Network ID) and the last address is the broadcast address for that subnet. The first and last addresses are not considered usable subnet addresses.

The following are the **ranges** (**increments**) for our Class C network in this example:

	Subnet Address	First Host Address	Last Host Address	Broadcast Address
0	~~192.62.123.0~~	~~192.62.123.1~~	~~192.62.123.14~~	~~192.62.123.15~~
1	192.62.123.16	192.62.123.17	192.62.123.30	192.62.123.31
2	192.62.123.32	192.62.123.33	192.62.123.46	192.62.123.47
3	192.62.123.48	192.62.123.49	192.62.123.62	192.62.123.63
4	192.62.123.64	192.62.123.65	192.62.123.78	192.62.123.79
5	192.62.123.80	192.62.123.81	192.62.123.94	192.62.123.95
6	192.62.123.96	192.62.123.97	192.62.123.110	192.62.123.111
7	192.62.123.112	192.62.123.113	192.62.123.126	192.62.123.127
8	192.62.123.128	192.62.123.129	192.62.123.142	192.62.123.143
9	192.62.123.144	192.62.123.145	192.62.123.158	192.62.123.159
10	192.62.123.160	192.62.123.161	192.62.123.175	192.62.123.175
11	192.62.123.176	192.62.123.177	192.62.123.190	192.62.123.191
12	192.62.123.192	192.62.123.193	192.62.123.206	192.62.123.207
13	192.62.123.208	192.62.123.209	192.62.123.222	192.62.123.223
14	192.62.123.224	192.62.123.225	192.62.123.238	192.62.123.239
~~15~~	~~192.62.123.240~~	~~192.62.123.241~~	~~192.62.123.254~~	~~192.62.123.255~~

You will notice that one other range could have been added, but the last range is always invalid because the subnet address cannot be made of all binary 1s (end in 255). Just remember to stop your ranges when you reach a subnet address that is the same as the subnet mask.

This chart also produces one other pertinent number in addition to the number of subnets. You guessed it, the **number of hosts per subnet**.

Once again, use the formula **2^N-2**. This time however, N is found a little differently. To find the number of hosts per subnet, N is now the number of bits **to the right of the subnet mask value**, instead of the left. Don't forget the other octets as well, if you are working on a Class A or B network.

Example… 240 yields:

```
      1,048,574 hosts per subnet on a Class A Network (2^20-2)
          4,094 hosts per subnet on a Class B Network (2^12-2)
             14 hosts per subnet on a Class C Network (2^ 4-2)
```

Here's another example, this time using a **Class B** network address. All of the same rules apply, but calculating the ranges is a little more confusing because you have **two octets** to contend with.

First, we are assigned a Class B Network ID: **172.190.0.0**

What subnet mask will you need for 10 subnets?

Once again, a quick look at our chart indicates 224 will only give you 6 subnets, but 240 yields you 14. The answer: **255.255.240.0**

All bits to the right of 240, including the other 8 bits for the last octet, make up the **Host ID**. The 240 and everything to the left of that make up the **Network ID**.

Now, let's look at the **ranges**, or **increments**. Look above 240 and you see 16. As in the previous example, this is the **incrementing range** and the beginning of the first range, but a ".0" needs to be added for the last octet (e.g. 16.0).

The next subnet address will be 16.0 + 16.0, or 32.0. The next subnet address will be 32.0 + 16.0, or 48.0, and so on. The first address in each range is the address for the subnet itself and the last address is the broadcast address for that subnet, neither of which can be a useable subnet host address. That rule doesn't change, but remember that the other octet is now included:

	Subnet Address	First Host Address	Last Host Address	Broadcast Address
0	~~172.190.0.0~~	~~172.190.0.1~~	~~172.190.15.254~~	~~172.190.15.255~~
1	172.190.16.0	172.190.16.1	172.190.31.254	172.190.31.255
2	172.190.32.0	172.190.32.1	172.190.47.254	172.190.47.255
3	172.190.48.0	172.190.48.1	172.190.63.254	172.190.63.255
4	172.190.64.0	172.190.64.1	172.190.79.254	172.190.79.255
5	172.190.80.0	172.190.80.1	172.190.95.254	172.190.95.255
6	172.190.96.0	172.190.96.1	172.190.111.254	172.190.111.255
7	172.190.112.0	172.190.112.1	172.190.127.254	172.190.127.255
8	172.190.128.0	172.190.128.1	172.190.143.254	172.190.143.255
9	172.190.144.0	172.190.144.1	172.190.159.254	172.190.159.255
10	172.190.160.0	172.190.160.1	172.190.175.254	172.190.175.255
11	172.190.176.0	172.190.176.1	172.190.191.254	172.190.191.255
12	172.190.192.0	172.190.192.1	172.190.207.254	172.190.207.255
13	172.190.208.0	172.190.208.1	172.190.223.254	172.190.223.255
14	172.190.224.0	172.190.224.1	172.190.239.254	172.190.239.255
15	~~172.190.240.0~~	~~172.190.240.1~~	~~172.190.255.254~~	~~172.190.255.255~~

Another tip on **subnet ranges**… you can use the **"Magic Number"** theory by subtracting the "non-255" number in the subnet mask from 256 to get the **incremental value**. In the examples above, if you subtract 240 from 256, the incremental value is 16!

Subnetting 1 2 3
By Adam Chee W.S

Ever get stressed out because you know that there would be subnetting question(s) in the next exam you are taking and that these questions easily take up 10 to 20 minutes of your precious exam time? What if there is more than one question?

The process of converting the subnet to binary and decimal can drive the unfamiliar insane, not to mention the waste of precious time and brain power which can be utilized for other areas of exam preparation.

Let's take a look at a shortcut method that will cut down the time needed to answer these questions without the need for a calculator.

Subnet Basics:

This article assumes that you know how to perform subnetting in the traditional method but it is important to stress that there are only 3 classes of usable IP addresses which are

Class	Range	Subnet mask	Host bit	Subnet
Class A	1 - 126 (127 is reserved for loopback)	255.0.0.0	24	8
Class B	128 - 191	255.255.0.0	16	16
Class C	192 - 223	255.255.255.0	8	24

You must understand and remember this table well in order to master the shortcut.
Note: You must borrow at least 2 bits and must leave at least 2 bits

The 'Subnet Table'

Bits Borrowed (N)	1	2	3	4	5	6	7	8
Bit Value	128	64	32	16	8	4	2	1
Subnet Mask	128	192	224	240	248	252	254	255
Number of Subnets ((2^N)-2)	0	2	6	14	30	62	126	254

If using the (2^N) method as defined in RFC 1878, your table would look like this instead.

Bits Borrowed (N)	1	2	3	4	5	6	7	8
Bit Value	128	64	32	16	8	4	2	1
Subnet Mask	128	192	224	240	248	252	254	255
Number of Subnets (2^N)	2	4	8	16	32	64	128	256

The 'subnet table' is commonly seen in lecture notes or certification guides, but what the author/lecturer did not tell you is how to derive this table on the fly. Its actually quite simple, lets look at it line by line.

1) Bits borrowed, this is the easy one, just remember that the table consists of only 8 columns.
2) Bits Value, remember by heart that the first value starts with 128 and the subsequent values are divided by two.
3) Subnet Mask, this line tells you what the subnet mask would be, to get the figures, add up the corresponding **bits value** and all of the values prior to it.
128 + 0 (there is no prior value) =128
128 + 64 = 192
192 + 32 = 224
224 + 16 = 240
240 + 8 = 248
248 + 4 = 252
252 + 2 + 254
254 + 1 + 255
4) Number of Subnets, tells you how many subnet you'll get if you use the subnet mask. Just look at the corresponding N value at the top and you can derive the figures.

Once you understand how to derive the 'subnet table', spend some time practicing. I would advise you to draw out the table once you are in the exam room (before starting the actual exam) it will take you less than a minute.

How to tackle the questions
There are only a few different ways that Microsoft or Cisco can phrase their questions, lets take a look at some examples,

Question Type 1:
If you are to determine the subnet mask based on a number of hosts and an IP address
Example:
You are assigned an IP address of 172.30.0.0 and you need 1000 hosts on your network, what is your subnet mask.

Step one: Determine the number of bits needed for the hosts.
In this scenario, we need ten bits as $2\char`^10 = 1024$ (the question asks for 1000 hosts only)

Step two: Determine the number of bits left for the subnet.
32 - (number of bits needed for the host) which is 32-10 = 22 bits

Step three: Determine the number of bits actually borrowed.
We take the number of bits left for the subnet and minus as many 8s as possible as each 8 represents 1 octal. Therefore 22 - 8 - 8 = 6 bits were borrowed

Bits Borrowed (N)	1	2	3	4	5	6	7	8
Bits Value	128	64	32	16	8	4	2	1
Subnet Mask	128	192	224	240	248	252	254	255
Number of Subnets (2N-2)	0	2	6	14	30	62	126	254

With reference to the subnet table, 6 bits would have a subnet of 255.255.252.0 . Take note that a total of two 8s were subtracted off, therefore the first two octal would be 255.255.x.x and the 3rd octal was 6 bits borrowed which leaves with 255.255.252.x.
Simple?

Question Type 2:
If you were given an IP address of 172.30.0.0 and you need 15 subnets

Bits Borrowed (N)	1	2	3	4	5	6	7	8
Bits Value	128	64	32	16	8	4	2	1
Subnet Mask	128	192	224	240	248	252	254	255
Number of Subnets (2N-2)	0	2	6	14	30	62	126	254

With reference to the subnet table, the subnet mask should be 255.255.248.0. 172.30.0.0 is a Class B address and the subnet should be 255.255.0.0.

Question Type 3:
You are assigned an IP address of 172.30.0.0 and you need 55 subnets, how many hosts do you have per subnet?

Step One: Determine the number of bits used for the subnet.

Bits Borrowed (N)	1	2	3	4	5	6	7	8
Bits Value	128	64	32	16	8	4	2	1
Subnet Mask	128	192	224	240	248	252	254	255
Number of Subnets (2N-2)	0	2	6	14	30	62	126	254

According to the chart, the closest match to 55 subnet would be 62 and therefore, the number of bits borrowed for the subnet is 6. Since 172.30.0.0 is a Class B, we would need to add another 16 bits to the 6 making it 22 bits in total.

Step Two: Determine the number of bits used for the host.
Number of bits used for the hosts is 32 - (number of bits used for the subnet) which is 22 = 10 bits.
$2^{10}-2 = 1022$, therefore there are a total of 1022 usable hosts in each subnet.

The key to mastering this shortcut is the same as with any other mathematical question - practice.

Good luck on your next exam

By Adam Chee W.S. (MCP ID 2915249)
http://www.adamchee.cjb.net

Exercises:

1. How many usable subnets (not theoretical) do you have when using a subnet mask of 255.255.255.240 on Network ID 201.114.168?

2. You are given Network ID 222.72.157, with a subnet mask of 255.255.255.248 to setup. How many subnets and hosts will you have?

3. You are assigned a Network ID of 198.162.10 and asked to configure the network to provide at least six useable subnets with at least 25 hosts on each subnet. What is the BEGINNING IP address of the LAST useable subnet in the network?

4. How many useable hosts are on each subnet when the Network ID is 199.215.210 and the subnet mask is 255.255.255.252?

5. You are given Network ID 190.90, with a subnet mask of 255.255.192.0 to setup. What are the high-order bits (Leading Bit Values) for this network?

6. You are assigned a Network ID of 162.160 and asked to configure the network to provide at least 60 useable subnets? What would be the subnet mask for this network?

7. How many useable hosts are on each subnet when the Network ID is 150.150 and the subnet mask is 255.255.192.0?

8. You are assigned a Network ID of 145.19 and asked to configure the network to provide at least 100 useable subnets with at 500 hosts on each subnet. What is the ENDING IP address of the EIGHTH useable subnet in the network?

9. You are a private contractor hired by the large company to setup the network for their enterprise. The Network ID is 33 and you need at least 125 subnets in their large network with at least 125,000 hosts on each of the subnets. What would be the subnet mask for this network?

10. You are given Network ID 55.0.0.0, with a subnet mask of 255.240.0.0 to setup. How many subnets and hosts will you have?

The Answer Key for these exercises is available at <u>EasySubnetting.com</u>

Subnet Worksheet

	Network Number:	
	Subnet Mask:	
	Number of Useable Subnets:	
	Number of Useable Hosts:	
	Increment:	

	Subnet Address	First Host Address	Last Host Address	Broadcast Address
0				
1				
2				
3				
4				
5				
6				
7				
8				
9				
10				
11				
12				
13				
14				
15				
16				
17				
18				
19				
20				
21				
22				
23				
24				