
cisco

APPENDIX D

Subnetting Practice

This appendix includes several sets of problems related to IP addressing and subnetting. The answer sections include not only the right answers, but also explanations of how the answers were found using the processes covered in the book. In particular, the answers show the steps as explained in Chapter 12, "IP Addressing and Subnetting," of the *CCENT/CCNA ICND1 Official Exam Certification Guide*. (Chapter 12 is also included in the *CCNA ICND2 Official Exam Certification Guide*, on the CD-ROM, as Appendix H.) For additional help, Appendix E, "Subnetting Reference Pages," summarizes the steps in individual reference pages (RP), so that you do not have to look around in ICND1 Chapter 12 (or ICND2 Appendix H) to find the right steps to solve a particular problem.

Appendix D, "Subnetting Practice," is identical in the *CCENT/CCNA ICND1 Official Exam Certification Guide* and the *CCNA ICND2 Official Exam Certification Guide*. Likewise, Appendix E, "Subnetting Reference Pages," is also identical in each of these two books. If you own both books, you can use either copy of these appendixes as you study. If you own only the *CCENT/CCNA ICND1 Official Exam Certification Guide*, just ignore the last problem set (Problem Set 8), because the material covered in that problem set is not included in the ICND1 exam. If you own only the *CCNA ICND2 Official Exam Certification Guide*, you should be able to find the answers to all the questions in this appendix in that book, but if you want to read more explanations about the processes, you will need to refer to Appendix H on the CD-ROM, which is a copy of *ICND1* Chapter 12, "IP Addressing and Subnetting."

Table D-1 lists the problem sets in this appendix. The table also notes which subnetting reference page or pages from Appendix E are used to find the answers for each set of problems.

Table D-1 *Problem Sets in this Appendix, and Corresponding Subnetting Reference Pages*

Problem Set	Description	RPs Used
1	Converting subnet masks from dotted decimal to prefix format, and vice versa	RP-1A, RP-1B
2	Basic interpretation of the address class, finding the network number and network broadcast address	RP-2
3	Interpreting an existing mask to find the number of subnets, number of hosts per subnet, and number of network, subnet, and host bits	RP-3A, RP-3B
4	Choosing the correct mask(s), given a set of requirements	RP-4
5	Given an IP address and mask, find the number of hosts in the subnet, number of subnets possible, subnet number, broadcast address, and range of usable IP addresses in the subnet	RP-5A, RP-5B, RP-5C, RP-6A, RP-6B, RP-6C
6	List all possible subnets of a classful network, assuming a static-length mask and fewer than 8 subnet bits	RP-7A
7	List all possible subnets of a classful network, assuming a static-length mask and more than 8 subnet bits	RP-7B
8	Find new subnet numbers to use for new subnets in an existing network, assuming VLSM	RP-8

Problem Sets 1–8

Problem Set 1: Converting Between Mask Formats

Problem Set 1 requires you to convert dotted decimal subnet masks to prefix format, and vice versa. To do so, feel free to use the processes described in Chapter 12 of the *CCENT/CCNA ICND1 Official Exam Certification Guide* (Appendix H in the *CCNA ICND2 Official Exam Certification Guide*), or use the summarized processes listed in Appendix E, RP-1A and RP-1B.

Convert each of the following masks to the other mask format:

1. 255.240.0.0
2. 255.255.192.0
3. 255.255.255.224
4. 255.254.0.0
5. 255.255.248.0
6. /30
7. /25
8. /11
9. /22
10. /24

Problem Set 2: Analyzing Unsubnetted IP Addresses

Problem Set 2 requires that you determine a few basic facts about a network, given an IP address and an assumption that subnetting is not used in that network. To do so, refer to the processes described in Chapter 12 of the *CCENT/CCNA ICND1 Official Exam Certification Guide* (Appendix H in the *CCNA ICND2 Official Exam Certification Guide*), or use the summarized processes listed in Appendix E, RP-2.

In particular, you should identify the following information:

- The class of the address
- The number of octets in the network part of the address

- The number of octets in the host part of the address
- The network number
- The network broadcast address

Find all of these facts for the following IP addresses:

1. 10.55.44.3
2. 128.77.6.7
3. 192.168.76.54
4. 190.190.190.190
5. 9.1.1.1
6. 200.1.1.1

Problem Set 3: Interpreting Existing Subnet Masks

Problem Set 3 lists problems that require you to analyze an existing IP address and mask to determine the number of network, subnet, and host bits. From that, you should calculate the number of subnets possible when using the listed mask in the class of network shown in the problem, as well as the number of possible host addresses in each subnet. In short, your task is to complete Table D-2.

Table D-2 *Problem Set 3*

Problem Number	Problem	Network Bits	Subnet Bits	Host Bits	Number of Subnets in Network	Number of Hosts per Subnet
1	10.66.5.99, 255.255.254.0					
2	172.16.203.42, 255.255.252.0					
3	192.168.55.55, 255.255.255.224					
4	10.22.55.87/30					
5	172.30.40.166/26					
6	192.168.203.18/29					

Note that for the purposes of this exercise, you can assume that the two special subnets in each network, the zero subnet and broadcast subnet, are allowed to be used.

To find this information, you can use processes explained in Chapter 12 of the *CCENT/CCNA ICND1 Official Exam Certification Guide* (Appendix H in the *CCNA ICND2 Official Exam Certification Guide*), or refer to the summarized version of those processes in Appendix E, RP-3A (binary process) and RP-3B (decimal process).

Problem Set 4: Choosing Subnet Masks

Problem Set 4 starts with a short set of requirements regarding how a particular classful network should be subnetted. The requirements include the classful network, the number of subnets the design must support, and the number of hosts in each subnet. For each problem, supply the following information:

- The minimum number of subnet and host bits needed in the mask to support the design requirements
- The dotted decimal format mask(s) that meet the requirements
- The mask you would choose if the problem said to maximize the number of subnets
- The mask you would choose if the problem said to maximize the number of hosts per subnet

To find this information, you can refer to Chapter 12 of the *CCENT/CCNA ICND1 Official Exam Certification Guide* (Appendix H in the *CCNA ICND2 Official Exam Certification Guide*), or refer to the summarized version of those processes in Appendix E, RP-4. Also note that you should assume that the two special subnets in each network, the zero subnet and broadcast subnet, are allowed to be used for these questions.

Find the key facts for these sets of requirements:

1. Network 10.0.0.0, need 50 subnets, need 200 hosts/subnet
2. Network 172.32.0.0, need 125 subnets, need 125 hosts/subnet
3. Network 192.168.44.0, need 15 subnets, need 6 hosts/subnet
4. Network 10.0.0.0, need 300 subnets, need 500 hosts/subnet
5. Network 172.32.0.0, need 500 subnets, need 15 hosts/subnet
6. Network 172.16.0.0, need 2000 subnets, need 2 hosts/subnet

Problem Set 5: Analyzing an Address in an Existing Subnet

Problem Set 5 asks you to find a wide variety of information about the subnet in which an IP address resides. Each problem supplies an IP address and a subnet mask, from which you should find the following information:

- Size of the network part of the address
- Size of the subnet part of the address
- Size of the host part of the address
- The number of hosts per subnet
- The number of subnets in this network
- The subnet number
- The broadcast address
- The range of valid IP addresses in this network

To find these facts, you can use any of the processes explained in Chapter 12 of the *CCENT/CCNA ICND1 Official Exam Certification Guide* (Appendix H in the *CCNA ICND2 Official Exam Certification Guide*), and you can refer to all the Appendix E reference pages that begin with "RP-3," "RP-5," or "RP-6."

Solve for the following problems:

1. 10.180.10.18, mask 255.192.0.0
2. 10.200.10.18, mask 255.224.0.0
3. 10.100.18.18, mask 255.240.0.0
4. 10.100.18.18, mask 255.248.0.0
5. 10.150.200.200, mask 255.252.0.0
6. 10.150.200.200, mask 255.254.0.0
7. 10.220.100.18, mask 255.255.0.0
8. 10.220.100.18, mask 255.255.128.0
9. 172.31.100.100, mask 255.255.192.0
10. 172.31.100.100, mask 255.255.224.0

11. 172.31.200.10, mask 255.255.240.0
12. 172.31.200.10, mask 255.255.248.0
13. 172.31.50.50, mask 255.255.252.0
14. 172.31.50.50, mask 255.255.254.0
15. 172.31.140.14, mask 255.255.255.0
16. 172.31.140.14, mask 255.255.255.128
17. 192.168.15.150, mask 255.255.255.192
18. 192.168.15.150, mask 255.255.255.224
19. 192.168.100.100, mask 255.255.255.240
20. 192.168.100.100, mask 255.255.255.248
21. 192.168.15.230, mask 255.255.255.252
22. 10.1.1.1, mask 255.248.0.0
23. 172.16.1.200, mask 255.255.240.0
24. 172.16.0.200, mask 255.255.255.192
25. 10.1.1.1, mask 255.0.0.0

Problem Set 6: Listing All Subnets of a Network (Fewer Than 8 Subnet Bits)

The problems in Problem Set 6 supply a classful network number and a mask. The mask is used throughout the network—in other words, static-length subnet masking (SLSM) is used. For each problem in this problem set, you should supply the following information:

- All subnet numbers
- The subnet that is the zero subnet
- The subnet that is the broadcast subnet

To find this information, you can use processes explained in Chapter 12 of the *CCENT/CCNA ICND1 Official Exam Certification Guide* (Appendix H in the *ICND2 Official Exam Certification Guide*), or refer to the summarized version of those processes in Appendix E, RP-7A.

The problems, which consist of a classful network and static-length mask, are as follows:

1. 172.32.0.0/22
2. 200.1.2.0/28
3. 10.0.0.0/15

Problem Set 7: Listing All Subnets of a Network (At Least 8 Subnet Bits)

The problems in Problem Set 7 begin by supplying a classful network number and a mask. The mask is used throughout the network—in other words, static-length subnet masking (SLSM) is used. For each problem, you should supply the following information:

- All subnet numbers
- The subnet that is the zero subnet
- The subnet that is the broadcast subnet

To find this information, you can use processes explained in Chapter 12 of the *CCENT/CCNA ICND1 Official Exam Certification Guide* (Appendix H in the *ICND2 Official Exam Certification Guide*), or refer to the summarized version of those processes in Appendix E, RP-7B.

The problems, which consist of a classful network and static-length mask, are as follows:

1. 172.32.0.0/25
2. 10.0.0.0/21
3. 172.20.0.0/24

Problem Set 8: Identifying a New Subnet to Add to a VLSM Design

> **NOTE** Only the *CCNA ICND2 Official Exam Certification Guide* explains the details needed to solve these problems; the *CCENT/CCNA ICND1 Official Exam Certification Guide* does not. The problems in Problem Set 8 are not applicable to the ICND1 exam.

The problems listed in Problem Set 8 begin with a working internetwork in which several subnets have been deployed, using different masks, along with a set of requirements for a new subnet. Your job is to identify the new subnet number that meets the requirements.

In each problem, you will be supplied the following:

- A list of existing subnet numbers as well as masks in prefix format.
- A statement of how many hosts must be supported in the new subnet. (You will need to then pick the mask with the least number of host bits that supports the stated number of hosts.)
- Whether to choose the numerically smallest, or largest, available new subnet number.

Although each problem simply requires that you list the correct new subnet number as the answer, you will need to do several other actions to make that choice. In particular, you will need to find the range of addresses in each existing subnet. You will need to pick a subnet mask that has the least number of host bits that supports the stated number of hosts. You will need to find the possible subnet numbers of that network, using that mask, and then pick a subnet that does not overlap with the existing subnets.

You can read more background information about how to attack this kind of problem in Chapter 5 of the *CCNA ICND2 Official Exam Certification Guide*. You can also see a summary of a process used to solve this problem in Appendix E, RP-8.

Choose New VLSM Subnet: Problem 1

Find the numerically smallest subnet number of network 172.30.0.0 that can be used for a new subnet, with the new subnet supporting up to 300 hosts. The following list shows all currently deployed subnets of network 172.30.0.0:

- 172.30.34.0/30
- 172.30.34.4/30
- 172.30.34.8/30
- 172.30.0.0/20
- 172.30.20.0/22
- 172.30.32.0/25
- 172.30.34.12/30
- 172.30.34.16/30

Choose New VLSM Subnet: Problem 2

Find the numerically largest subnet number of network 192.168.1.0 that can be used for a new subnet, with the new subnet supporting up to 13 hosts. The following list shows all currently deployed subnets of network 192.168.1.0:

- 192.168.1.192/26
- 192.168.1.64/30
- 192.168.1.72/30
- 192.168.1.76/30
- 192.168.1.128/26

Answers to Problem Sets 1–8

Answers to Problem Set 1

This section includes the answers to the ten problems listed in Problem Set 1. The answer section for each problem lists an explanation of how both the binary and decimal processes were used to find the answer.

Answer to Problem 1 in Problem Set 1

The answer is /12.

The binary process for converting the mask from dotted decimal format to prefix format is relatively simple. The only hard part is converting the dotted decimal number to binary. For reference, the process is as follows:

Step 1 Convert the dotted decimal mask to binary.

Step 2 Count the number of binary 1s in the 32-bit binary mask; this is the value of the prefix notation mask.

For problem 1, mask 255.240.0.0 converts to:

```
11111111 11110000 00000000 00000000
```

You can see from the binary number that it contains 12 binary 1s, so the prefix format of the mask will be /12.

You can find the exact same answer without converting decimal to binary by using the same steps as outlined in RP-2B in Appendix E. This process requires that you remember the nine decimal numbers that can be used in a subnet mask and their binary equivalents. Follow these steps:

Step 1 Start with a prefix value of 0.

Step 2 (1^{st} octet) Add 8 because the *first* mask octet of 255 includes eight binary 1s.

Step 2 (2^{nd} octet) Add 4 because the *second* mask octet of 240 includes four binary 1s.

Step 3 The resulting prefix is /12.

Answer to Problem 2 in Problem Set 1

The answer is /18.

For problem 2, mask 255.255.192.0 converts to:

```
11111111 11111111 11000000 00000000
```

You can see from the binary number that it contains 18 binary 1s, so the prefix format of the mask will be /18.

Using the decimal process found in Appendix E, RP-2B, follow these steps:

Step 1 Start with a prefix value of 0.

Step 2 (1st octet) Add 8 because the *first* mask octet of 255 includes eight binary 1s.

Step 2 (2nd octet) Add 8 because the *second* mask octet of 255 includes eight binary 1s.

Step 2 (3rd octet) Add 2 because the *third* mask octet of 192 includes two binary 1s.

Step 3 The resulting prefix is /18.

Answer to Problem 3 in Problem Set 1

The answer is /27.

For problem 3, mask 255.255.255.224 converts to:

```
11111111 11111111 11111111 11100000
```

You can see from the binary number that it contains 27 binary 1s, so the prefix format of the mask will be /27.

Using the decimal process found in Appendix E, RP-2B, follow these steps:

Step 1 Start with a prefix value of 0.

Step 2 (1st octet) Add 8 because the *first* mask octet of 255 includes eight binary 1s.

Step 2 (2nd octet) Add 8 because the *second* mask octet of 255 includes eight binary 1s.

Step 2 (3rd octet) Add 8 because the *third* mask octet of 255 includes eight binary 1s.

Step 2 (4th octet) Add 3 because the *fourth* mask octet of 224 includes three binary 1s.

Step 3 The resulting prefix is /27.

Answer to Problem 4 in Problem Set 1

The answer is /15.

For problem 4, mask 255.254.0.0 converts to:

```
11111111 11111110 00000000 00000000
```

You can see from the binary number that it contains 15 binary 1s, so the prefix format of the mask will be /15.

Using the decimal process found in Appendix E, RP-2B, follow these steps:

Step 1 Start with a prefix value of 0.

Step 2 (1st octet) Add 8 because the *first* mask octet of 255 includes eight binary 1s.

Step 2 (2nd octet) Add 7 because the *second* mask octet of 254 includes seven binary 1s.

Step 3 The resulting prefix is /15.

Answer to Problem 5 in Problem Set 1

The answer is /21.

For problem 5, mask 255.255.248.0 converts to:

```
11111111 11111111 11111000 00000000
```

You can see from the binary number that it contains 21 binary 1s, so the prefix format of the mask will be /21.

Using the decimal process found in Appendix E, RP-2B, follow these steps:

Step 1 Start with a prefix value of 0.

Step 2 (1st octet) Add 8 because the *first* mask octet of 255 includes eight binary 1s.

Step 2 (2nd octet) Add 8 because the *second* mask octet of 255 includes eight binary 1s.

Step 2 (3rd octet) Add 5 because the *third* mask octet of 248 includes five binary 1s.

Step 3 The resulting prefix is /21.

Answer to Problem 6 in Problem Set 1

The answer is 255.255.255.252.

The binary process (according to Appendix E, RP-2A) for converting the prefix version of the mask to dotted decimal is straightforward, but again requires some binary math. For reference, the process runs like this:

Step 1 Write down *x* binary 1s, where *x* is the value listed in the prefix version of the mask.

Step 2 Write down binary 0s after the binary 1s until the combined 1s and 0s form a 32-bit number.

Step 3 Convert this binary number, 8 bits at a time, to decimal, to create a dotted decimal number; this value is the dotted decimal version of the subnet mask.

For problem 6, with a prefix of /30, you start at Step 1 by writing down 30 binary 1s, as shown here:

```
11111111 11111111 11111111 111111
```

At Step 2, you add binary 0s until you have 32 total bits, as shown next:

```
11111111 11111111 11111111 11111100
```

The only remaining work is to convert this 32-bit number to decimal, remembering that the conversion works with 8 bits at a time.

The decimal process is a bit more detailed, but again avoids binary math. For reference, Appendix E, RP-2B, defines this process, which represents the prefix length as the variable *x*:

Step 1 Divide *x* by 8 (*x*/8), noting the number of times 8 fully goes into *x* (the dividend, represented as a *d*), and the number left over (the remainder, represented as an *r*).

Step 2 Write down *d* octets of value 255.

Step 3 For the next octet, find the decimal number that begins with *r* binary 1s, followed by all binary 0s. (This step requires the memorization of the nine decimal numbers allowed in a subnet mask, and their binary equivalents.)

Step 4 For any remaining octets, write down a decimal 0.

For instance, in this case, follow these steps:

Step 1 The prefix length (30), divided by 8, gives a dividend of 3 and a remainder of 6.

Step 2 Because the dividend is 3, begin the mask with 3 octets of 255.

Step 3 Because the remainder is 6, and 11111100 is equal to decimal 252, write down 252 for the next octet.

Step 4 (No need for Step 4 in this case.)

The resulting mask is 255.255.255.252.

Answer to Problem 7 in Problem Set 1

The answer is 255.255.255.128.

For problem 7, with a prefix of /25, you start at Step 1 by writing down 25 binary 1s, as shown here:

```
11111111 11111111 11111111 1
```

At Step 2, you add binary 0s until you have 32 total bits, as shown next:

```
11111111 11111111 11111111 10000000
```

The only remaining work is to convert this 32-bit number to decimal, remembering that the conversion works with 8 bits at a time.

The decimal process is a bit more detailed, but again avoids binary math. For instance, in this case, follow these steps:

Step 1 The prefix length (25), divided by 8, gives a dividend of 3 and a remainder of 1.

Step 2 Because the dividend is 3, begin the mask with 3 octets of 255.

Step 3 Because the remainder is 1, and 10000000 is equal to decimal 128, write down 128 for the next octet.

Step 4 (No need for Step 4 in this case.)

The resulting mask is 255.255.255.128.

Answer to Problem 8 in Problem Set 1

The answer is 255.224.0.0.

For problem 8, with a prefix of /11, you start at Step 1 by writing down 11 binary 1s, as shown here:

```
11111111 111
```

At Step 2, you add binary 0s until you have 32 total bits, as shown next:

```
11111111 11100000 00000000 00000000
```

The only remaining work is to convert this 32-bit number to decimal, remembering that the conversion works with 8 bits at a time.

The decimal process is a bit more detailed, but again avoids binary math. For instance, in this case, follow these steps:

Step 1 The prefix length (11), divided by 8, gives a dividend of 1 and a remainder of 3.

Step 2 Because the dividend is 1, begin the mask with 1 octet of 255.

Step 3 Because the remainder is 3, and 11100000 is equal to decimal 224, write down 224 for the next octet.

Step 4 Nothing has been written down yet for the last two octets, so write down decimal 0 for each of these last two octets.

The resulting mask is 255.224.0.0.

Answer to Problem 9 in Problem Set 1

The answer is 255.255.252.0.

For problem 9, with a prefix of /22, you start at Step 1 by writing down 22 binary 1s, as shown here:

```
11111111 11111111 111111
```

At Step 2, you add binary 0s until you have 32 total bits, as shown next:

```
11111111 11111111 11111100 00000000
```

The only remaining work is to convert this 32-bit number to decimal, remembering that the conversion works with 8 bits at a time.

The decimal process is a bit more detailed, but again avoids binary math. For instance, in this case, follow these steps:

Step 1 The prefix length (22), divided by 8, gives a dividend of 2 and a remainder of 6.

Step 2 Because the dividend is 2, begin the mask with two octets of 255.

Step 3 Because the remainder is 6, and 11111100 is equal to decimal 252, write down 252 for the next octet.

Step 4 Nothing has been written down yet for the last octet, so write down decimal 0 for this last octet.

The resulting mask is 255.255.252.0.

Answer to Problem 10 in Problem Set 1

The answer is 255.255.255.0.

For problem 10, with a prefix of /24, you start at Step 1 by writing down 24 binary 1s, as shown here:

```
11111111 11111111 11111111
```

At Step 2, you add binary 0s until you have 32 total bits, as shown next:

```
11111111 11111111 11111111 00000000
```

The only remaining work is to convert this 32-bit number to decimal, remembering that the conversion works with 8 bits at a time.

The decimal process is a bit more detailed, but again avoids binary math. For instance, in this case, follow these steps:

Step 1 The prefix length (24), divided by 8, gives a dividend of 3 and a remainder of 0.

Step 2 Because the dividend is 3, begin the mask with 3 octets of 255.

Step 3 Because the remainder is 0, and 00000000 is equal to decimal 0, write down 0 for the next octet.

Step 4 (No need for Step 4 in this case.)

The resulting mask is 255.255.255.0.

Answers to Problem Set 2

This section includes the answers to problems 1–6 listed in Problem Set 2. The process to answer these problems is relatively basic, so this section reviews the overall process and then lists the answers to problems 1–6.

The process starts by examining the first octet of the IP address:

- If the first octet of the IP address is a number between 1–126, inclusive, then the address is a Class A address.
- If the first octet of the IP address is a number between 128–191, inclusive, the address is a Class B address.
- If the first octet of the IP address is a number between 192–223, inclusive, the address is a Class C address.

When no subnetting is used:

- Class A addresses have 1 octet in the network part of the address, and 3 octets in the host part.
- Class B addresses have 2 octets each in the network and host part.
- Class C addresses have 3 octets in the network part, and 1 octet in the host part.

After determining the class and the number of network octets, you can easily find the network number and network broadcast address. To find the network number, copy the network octets of the IP address, and write down 0s for the host octets. To find the network broadcast address, copy the network octets of the IP address, and write down 255s for the host octets.

Table D-3 lists all six problems and their respective answers.

Table D-3 *Answers to Problem Set 2*

IP Address	Number of Network Octets	Number of Host Octets	Network Number	Network Broadcast Address
10.55.44.3	1	3	10.0.0.0	10.255.255.255
128.77.6.7	2	2	128.77.0.0	128.77.255.255
192.168.76.54	3	1	192.168.76.0	192.168.76.255
190.190.190.190	2	2	190.190.0.0	190.190.255.255
9.1.1.1	1	3	9.0.0.0	9.255.255.255
200.1.1.1	3	1	200.1.1.0	200.1.1.255

Answers to Problem Set 3

Table D-4 includes the answers to problems 1–6 listed in Problem Set 3. The paragraphs following the table provide explanations of each of the answers.

Table D-4 *Answers to Problem Set 3*

Problem number	Problem	Network Bits	Subnet Bits	Host Bits	Number of Subnets in Network	Number of Hosts per Subnet
1	10.66.5.99, 255.255.254.0	8	15	9	2^{15} = 32,768	$2^9 - 2$ = 510
2	172.16.203.42, 255.255.252.0	16	6	10	$2^6 = 64$	$2^{10} - 2$ = 1022
3	192.168.55.55, 255.255.255.224	24	3	5	$2^3 = 8$	$2^5 - 2 = 30$
4	10.22.55.87/30	8	22	2	2^{22} = 4,194,304	$2^2 - 2 = 2$
5	172.30.40.166/26	16	10	6	$2^{10} = 1024$	$2^6 - 2 = 62$
6	192.168.203.18/29	24	5	3	$2^5 = 32$	$2^3 - 2 = 6$

Answer to Problem 1 in Problem Set 3

Address 10.66.5.99 is in Class A network 10.0.0.0, meaning 8 network bits exist. Mask 255.255.254.0 converts to prefix /23, because the first two octets of value 255 represent 8 binary 1s, and the 254 in the third octet represents 7 binary 1s, for a total of 23 binary 1s. Therefore, the number of host bits is 32 – 23 = 9, leaving 15 subnet bits (32 – 8 network bits – 9 host bits = 15 subnet bits). The number of subnets in this Class A network, using mask 255.255.254.0, is 2^{15} = 32,768. The number of hosts per subnet is $2^9 - 2 = 510$.

Answer to Problem 2 in Problem Set 3

Address 172.16.203.42, mask 255.255.252.0, is in Class B network 172.16.0.0, meaning 16 network bits exist. Mask 255.255.252.0 converts to prefix /22, because the first two octets of value 255 represent 8 binary 1s, and the 252 in the third octet represents 6 binary 1s, for a total of 22 binary 1s. Therefore, the number of host bits is 32 – 22 = 10, leaving 6 subnet bits (32 – 16 network bits – 10 host bits = 6 subnet bits). The number of subnets in this Class B network, using mask 255.255.252.0, is $2^6 = 64$. The number of hosts per subnet is $2^{10} - 2 = 1022$.

Answer to Problem 3 in Problem Set 3

Address 192.168.55.55 is in Class C network 192.168.55.0, meaning 24 network bits exist. Mask 255.255.255.224 converts to prefix /27, because the first three octets of value 255 represent 8 binary 1s, and the 224 in the fourth octet represents 3 binary 1s, for a total of 27 binary 1s. Therefore, the number of host bits is 32 – 27 = 5, leaving 3 subnet bits (32 – 24 network bits – 5 host bits = 3 subnet bits). The number of subnets in this Class C network, using mask 255.255.255.224, is $2^3 = 8$. The number of hosts per subnet is $2^5 - 2 = 30$.

Answer to Problem 4 in Problem Set 3

Address 10.22.55.87 is in Class A network 10.0.0.0, meaning 8 network bits exist. The prefix format mask of /30 lets you calculate the number of host bits as 32 – prefix-length, in this case 32 – 30 = 2. This leaves 22 subnet bits, because 32 – 8 network bits – 2 host bits = 22 subnet bits. The number of subnets in this Class A network, using mask 255.255.255.252, is $2^{22} = 4{,}194{,}304$. The number of hosts per subnet is $2^2 - 2 = 2$. (Note that this mask is popularly used on serial links, which need only two IP addresses in a subnet.)

Answer to Problem 5 in Problem Set 3

Address 172.30.40.166 is in Class B network 172.30.0.0, meaning 16 network bits exist. The prefix format mask of /26 lets you calculate the number of host bits as 32 – prefix-length, in this case 32 – 26 = 6. This leaves 10 subnet bits, because 32 – 16 network bits – 6 host bits = 10 subnet bits. The number of subnets in this Class B network, using mask /26, is $2^{10} = 1024$. The number of hosts per subnet is $2^6 - 2 = 62$.

Answer to Problem 6 in Problem Set 3

Address 192.168.203.18 is in Class C network 192.168.203.0, meaning 24 network bits exist. The prefix format mask of /29 lets you calculate the number of host bits as 32 – prefix-length, in this case 32 – 29 = 3. This leaves 5 subnet bits, because 32 – 24 network bits – 3 host bits = 5 subnet bits. The number of subnets in this Class C network, using mask /29, is $2^5 = 32$. The number of hosts per subnet is $2^3 - 2 = 6$.

Answers to Problem Set 4

This section includes the answers to the six problems listed in Problem Set 4. The answer section for each problem explains how to use the process outlined in Chapter 4, and summarized in Appendix E, RP-4, to find the answers.

Answer to Problem 1 in Problem Set 4

Problem 1 shows a Class A network, with 8 network bits, with a minimum of 6 subnet bits and 8 host bits to meet the required number of subnets and hosts/subnet. The following masks all meet the requirements in this problem, with the masks that maximize the number of hosts/subnet and the number of subnets noted:

- 255.252.0.0 (maximizes the number of hosts per subnet)
- 255.254.0.0
- 255.255.0.0
- 255.255.128.0
- 255.255.192.0
- 255.255.224.0
- 255.255.240.0
- 255.255.248.0
- 255.255.252.0
- 255.255.254.0
- 255.255.255.0 (maximizes the number of subnets)

As for the process to find the answers, the following list explains the details:

Step 1 The question lists Class A network 10.0.0.0, so there are 8 network bits.

Step 2 The question states that 50 subnets are needed. A mask with 5 subnet bits supplies only 2^5 (32) subnets, but a mask with 6 subnet bits supplies 2^6 (64) subnets. So, the mask needs at least 6 subnet bits.

Step 3 The question states that 200 hosts are needed per subnet. A mask with 7 host bits supplies only $2^7 - 2$ (126) hosts per subnet, but a mask with 8 host bits supplies $2^8 - 2$ (254) hosts per subnet. So, the mask needs at least 8 host bits.

Step 4 With 8 network bits and a minimum of 6 subnet bits, at this step you should write down 14 consecutive binary 1s, as follows:

```
11111111 111111
```

Step 5 With a minimum of 8 host bits, for this step, starting on the right, write down eight binary 0s, ending as follows:

```
11111111 111111              00000000
```

Step 6 Obviously, several bit positions do not have a value yet, so the two substeps for Step 6 must be performed:

a. Represent the value with Xs in the wildcard positions, as follows:

```
11111111 111111XX XXXXXXXX 00000000
```

b. Substitute all binary 0s for the Xs to find one mask; then substitute a binary 1, with the rest binary 0s, to find the next mask; and so on until you substitute all binary 1s for the Xs, as follows:

```
11111111 11111100 00000000 00000000
11111111 11111110 00000000 00000000
11111111 11111111 00000000 00000000
11111111 11111111 10000000 00000000
11111111 11111111 11000000 00000000
11111111 11111111 11100000 00000000
11111111 11111111 11110000 00000000
11111111 11111111 11111000 00000000
11111111 11111111 11111100 00000000
11111111 11111111 11111110 00000000
11111111 11111111 11111111 00000000
```

Step 7 Convert each number back to dotted decimal or prefix notation as required by the question. The dotted decimal answers are listed at the beginning of this section. The prefix format masks are /14, /15, /16, and so on, up through /24.

Step 8 To pick the mask that maximizes the number of subnets, pick the mask with the most binary 1s from the list at Step 6b—namely, /24 (255.255.255.0). The mask that maximizes the number of hosts per subnet is on the top of the list, with the largest number of binary 0s—namely, /14 (255.252.0.0).

Answer to Problem 2 in Problem Set 4

Problem 2 shows a Class B network, with 16 network bits, with a minimum of 7 subnet bits and 7 host bits to meet the required number of subnets and hosts/subnet. The following masks all meet the requirements in this problem, with the masks that maximize the number of hosts/subnet and the number of subnets noted:

- 255.255.254.0 (maximizes the number of hosts/subnet)
- 255.255.255.0
- 255.255.255.128 (maximizes the number of subnets)

As for the process to find the answers, the following list explains the details:

Step 1 The question lists Class B network 172.32.0.0, so there are 16 network bits.

Step 2 The question states that 125 subnets are needed. A mask with 6 subnet bits supplies only 2^6 (64) subnets, but a mask with 7 subnet bits supplies 2^7 (128) subnets. So, the mask needs at least 7 subnet bits.

Step 3 The question states that 125 hosts are needed per subnet. A mask with 6 host bits supplies only $2^6 - 2$ (62) hosts per subnet, but a mask with 7 host bits supplies $2^7 - 2$ (126) hosts per subnet. So, the mask needs at least 7 host bits.

Step 4 With 16 network bits and a minimum of 7 subnet bits, at this step you should write down 23 consecutive binary 1s, as follows:

```
11111111 11111111 1111111
```

Step 5 With a minimum of 7 host bits, for this step, starting on the right, write down seven binary 0s, ending as follows:

```
11111111 11111111 1111111   0000000
```

Step 6 Obviously, several bit positions do not have a value yet, so the two substeps for Step 6 must be performed:

a. Represent the value with Xs in the wildcard positions, as follows:

```
11111111 11111111 1111111X X0000000
```

b. Substitute all binary 0s for the Xs to find one mask; then substitute a binary 1, with the rest binary 0s, to find the next mask; and so on until you substitute all binary 1s for the Xs, as follows:

```
11111111 11111111 11111110 00000000
11111111 11111111 11111111 00000000
11111111 11111111 11111111 10000000
```

Step 7 Convert each number back to dotted decimal or prefix notation as required by the question. The dotted decimal answers are listed at the beginning of this section. The prefix format masks are /23, /24, and /25.

Step 8 To pick the mask that maximizes the number of subnets, pick the mask with the most binary 1s from the list at Step 6b—namely, /25 (255.255.255.128). The mask that maximizes the number of hosts per subnet is on the top of the list, with the largest number of binary 0s—namely, /23 (255.255.254.0).

Answer to Problem 3 in Problem Set 4

Problem 3 shows a Class C network, with 24 network bits, with a minimum of 4 subnet bits and 3 host bits to meet the required number of subnets and hosts/subnet. The following masks all meet the requirements in this problem, with the masks that maximize the number of hosts/subnet and the number of subnets noted:

- 255.255.255.240 (maximizes the number of hosts/subnet)
- 255.255.255.248 (maximizes the number of subnets)

As for the process to find the answers, the following list explains the details:

Step 1 The question lists Class C network 192.168.44.0, so there are 24 network bits.

Step 2 The question states that 15 subnets are needed. A mask with 3 subnet bits supplies only 2^3 (8) subnets, but a mask with 4 subnet bits supplies 2^4 (16) subnets. So, the mask needs at least 4 subnet bits.

Step 3 The question states that 6 hosts are needed per subnet. A mask with 2 host bits supplies only $2^2 - 2$ (2) hosts per subnet, but a mask with 3 host bits supplies $2^3 - 2$ (6) hosts per subnet. So, the mask needs at least 3 host bits.

Step 4 With 24 network bits and a minimum of 4 subnet bits, at this step you should write down 28 consecutive binary 1s, as follows:

```
11111111 11111111 11111111 1111
```

Step 5 With a minimum of 3 host bits, for this step, starting on the right, write down three binary 0s, ending as follows:

```
11111111 11111111 11111111 1111 000
```

Step 6 One bit position does not have a value yet, so the two substeps for Step 6 must be performed:

a. Represent the value with Xs in the wildcard positions, as follows:

```
11111111 11111111 11111111 1111X000
```

b. With only one wildcard digit, find one mask by substituting a binary 0 for the X, and the other mask by substituting a binary 1 for the X:

```
11111111 11111111 11111111 11110000

11111111 11111111 11111111 11111000
```

Step 7 Convert each number back to dotted decimal or prefix notation as required by the question. The dotted decimal answers are listed at the beginning of this section. The prefix format masks are /28 and /29.

Step 8 To pick the mask that maximizes the number of subnets, pick the mask with the most binary 1s from the list at Step 6B—namely, /29 (255.255.255.248). The mask that maximizes the number of hosts per subnet is on the top of the list, with the largest number of binary 0s—namely, /28 (255.255.255.240).

Answer to Problem 4 in Problem Set 4

Problem 4 shows a Class A network, with 8 network bits, with a minimum of 9 subnet bits and 9 host bits to meet the required number of subnets and hosts/subnet. The following masks all meet the requirements in this problem, with the masks that maximize the number of hosts/subnet and the number of subnets noted:

- 255.255.128.0 (maximizes the number of hosts/subnet)
- 255.255.192.0
- 255.255.224.0
- 255.255.240.0
- 255.255.248.0
- 255.255.252.0
- 255.255.254.0 (maximizes the number of subnets)

As for the process to find the answers, the following list explains the details:

Step 1 The question lists Class A network 10.0.0.0, so there are 8 network bits.

Step 2 The question states that 300 subnets are needed. A mask with 8 subnet bits supplies only 2^8 (256) subnets, but a mask with 9 subnet bits supplies 2^9 (512) subnets. So, the mask needs at least 9 subnet bits.

Step 3 The question states that 500 hosts are needed per subnet. A mask with 8 host bits supplies only $2^8 - 2$ (254) hosts per subnet, but a mask with 9 host bits supplies $2^9 - 2$ (510) hosts per subnet. So, the mask needs at least 9 host bits.

Step 4 With 8 network bits and a minimum of 9 subnet bits, at this step you should write down 17 consecutive binary 1s, as follows:

```
11111111 11111111 1
```

Step 5 With a minimum of 9 host bits, for this step, starting on the right, write down nine binary 0s, ending as follows:

```
11111111 11111111 1      0 00000000
```

Step 6 Obviously, several bit positions do not have a value yet, so the two substeps for Step 6 must be performed:

a. Represent the value with Xs in the wildcard positions, as follows:

```
11111111 11111111 1XXXXXX0 00000000
```

b. Substitute all binary 0s for the Xs to find one mask; then substitute a binary 1, with the rest binary 0s, to find the next mask; and so on until you substitute all binary 1s for the Xs, as follows:

```
11111111 11111111 10000000 00000000
11111111 11111111 11000000 00000000
11111111 11111111 11100000 00000000
11111111 11111111 11110000 00000000
11111111 11111111 11111000 00000000
11111111 11111111 11111100 00000000
11111111 11111111 11111110 00000000
```

Step 7 Convert each number back to dotted decimal or prefix notation as required by the question. The dotted decimal answers are listed at the beginning of this section. The prefix format masks are /17, /18, /19, /20, /21, /22, and /23.

Step 8 To pick the mask that maximizes the number of subnets, pick the mask with the most binary 1s from the list at Step 6b—namely, /23 (255.255.254.0). The mask that maximizes the number of hosts per subnet is on the top of the list, with the largest number of binary 0s—namely, /17 (255.255.128.0).

Answer to Problem 5 in Problem Set 4

Problem 5 shows a Class B network, with 16 network bits, with a minimum of 9 subnet bits and 5 host bits to meet the required number of subnets and hosts/subnet. The following masks all meet the requirements in this problem, with the masks that maximize the number of hosts/subnet and the number of subnets noted:

- 255.255.255.128 (maximizes the number of hosts/subnet)
- 255.255.255.192
- 255.255.255.224 (maximizes the number of subnets)

As for the process to find the answers, the following list explains the details:

Step 1 The question lists Class B network 172.32.0.0, so there are 16 network bits.

Step 2 The question states that 500 subnets are needed. A mask with 8 subnet bits supplies only 2^8 (256) subnets, but a mask with 9 subnet bits supplies 2^9 (512) subnets. So, the mask needs at least 9 subnet bits.

Step 3 The question states that 15 hosts are needed per subnet. A mask with 4 host bits supplies only $2^4 - 2$ (14) hosts per subnet, but a mask with 5 host bits supplies $2^5 - 2$ (30) hosts per subnet. So, the mask needs at least 5 host bits.

Step 4 With 16 network bits and a minimum of 9 subnet bits, at this step you should write down 25 consecutive binary 1s, as follows:

```
11111111 11111111 11111111 1
```

Step 5 With a minimum of 5 host bits, for this step, starting on the right, write down five binary 0s, ending as follows:

```
11111111 11111111 11111111 1  00000
```

Step 6 Obviously, several bit positions do not have a value yet, so the two substeps for Step 6 must be performed:

a. Represent the value with Xs in the wildcard positions, as follows:

```
11111111 11111111 11111111 1XX00000
```

b. Substitute all binary 0s for the Xs to find one mask; then substitute a binary 1, with the rest binary 0s, to find the next mask; and so on until you substitute all binary 1s for the Xs, as follows:

```
11111111 11111111 11111111 10000000

11111111 11111111 11111111 11000000

11111111 11111111 11111111 11100000
```

Step 7 Convert each number back to dotted decimal or prefix notation as required by the question. The dotted decimal answers are listed at the beginning of this section. The prefix format masks are /25, /26, and /27.

Step 8 To pick the mask that maximizes the number of subnets, pick the mask with the most binary 1s from the list at Step 6b—namely, /27 (255.255.255.224). The mask that maximizes the number of hosts per subnet is on the top of the list, with the largest number of binary 0s—namely, /25 (255.255.255.128).

Answer to Problem 6 in Problem Set 4

Problem 6 shows a Class B network, with 16 network bits, with a minimum of 11 subnet bits and 2 host bits to meet the required number of subnets and hosts/subnet. The following masks all meet the requirements in this problem, with the masks that maximize the number of hosts/subnet and the number of subnets noted:

- 255.255.255.224 (maximizes the number of hosts/subnet)
- 255.255.255.240
- 255.255.255.248
- 255.255.255.252 (maximizes the number of subnets)

As for the process to find the answers, the following list explains the details:

Step 1 The question lists Class B network 172.16.0.0, so there are 16 network bits.

Step 2 The question states that 2000 subnets are needed. A mask with 10 subnet bits supplies only 2^{10} (1024) subnets, but a mask with 11 subnet bits supplies 2^{11} (2048) subnets. So, the mask needs at least 11 subnet bits.

Step 3 The question states that 2 hosts are needed per subnet. A mask with 2 host bits supplies $2^2 - 2$ (2) hosts per subnet. So, the mask needs at least 2 host bits.

Step 4 With 16 network bits and a minimum of 11 subnet bits, at this step you should write down 27 consecutive binary 1s, as follows:

```
11111111 11111111 11111111 111
```

Step 5 With a minimum of 2 host bits, for this step, starting on the right, write down two binary 0s, ending as follows:

```
11111111 11111111 11111111 111   00
```

Step 6 Obviously, several bit positions do not have a value yet, so the two substeps for Step 6 must be performed:

a. Represent the value with Xs in the wildcard positions, as follows:

```
11111111 11111111 11111111 111XXX00
```

b. Substitute all binary 0s for the Xs to find one mask; then substitute a binary 1, with the rest binary 0s, to find the next mask; and so on until you substitute all binary 1s for the Xs, as follows:

```
11111111 11111111 11111111 11100000

11111111 11111111 11111111 11110000

11111111 11111111 11111111 11111000

11111111 11111111 11111111 11111100
```

Step 7 Convert each number back to dotted decimal or prefix notation as required by the question. The dotted decimal answers are listed at the beginning of this section. The prefix format masks are /27, /28, /29, and /30.

Step 8 To pick the mask that maximizes the number of subnets, pick the mask with the most binary 1s from the list at Step 6b—namely, /30 (255.255.255.252). The mask that maximizes the number of hosts per subnet is on the top of the list, with the largest number of binary 0s—namely, /27 (255.255.255.224).

Answers to Problem Set 5

This section includes the answers to the 25 problems listed in Problem Set 5.

Answer to Problem 1 in Problem Set 5

The answers begin with the analysis of the three parts of the address, the number of hosts per subnet, and the number of subnets of this network using the stated mask, as outlined in Table D-5. The binary math for subnet and broadcast address calculation follows. The answer finishes with the easier mental calculations for the range of IP addresses in the subnet.

The processes used in the explanations to the answers in this section can be found in Chapter 12 of the *CCENT/CCNA ICND1 Official Exam Certification Guide* (Appendix H in the *CCNA ICND2 Official Exam Certification Guide*), as well as in the brief summary of the RP-3x, RP-5x, and RP-6x reference pages found in Appendix E.

Table D-5 *Question 1: Size of Network, Subnet, Host, Number of Subnets, Number of Hosts*

Item	Example	Rules to Remember
Address	10.180.10.18	—
Mask	255.192.0.0	—
Number of network bits	8	Always defined by Class A, B, C
Number of host bits	22	Always defined as number of binary 0s in mask
Number of subnet bits	2	32 – (network size + host size)
Number of subnets	$2^2 = 4$	$2^{\text{number-of-subnet-bits}}$
Number of hosts	$2^{22} - 2 = 4,194,302$	$2^{\text{number-of-host-bits}} - 2$

Table D-6 contains the important binary calculations for finding the subnet number and subnet broadcast address, as summarized in Appendix E, RP-5A. To calculate the subnet number, perform a Boolean AND on the address and mask. To find the broadcast address for this subnet, change all the host bits to binary 1s in the subnet number. The host bits are in **bold** print in the table.

Table D-6 *Question 1: Binary Calculation of Subnet and Broadcast Addresses*

Address	10.180.10.18	00001010 10**110100 00001010 00010010**
Mask	255.192.0.0	11111111 11**000000 00000000 00000000**
AND result (subnet number)	10.128.0.0	00001010 10**000000 00000000 00000000**
Change host to 1s (broadcast address)	10.191.255.255	00001010 10**111111 11111111 11111111**

To get the first valid IP address, just add 1 to the subnet number; to get the last valid IP address, just subtract 1 from the broadcast address. In this case:

10.128.0.1 through 10.191.255.254
10.128.0.0 + 1 = 10.128.0.1
10.191.255.255 – 1 = 10.191.255.254

Alternately, you can use the processes that only use decimal math to find the subnet and broadcast address. The key parts of the process are as follows:

- The interesting octet is the octet for which the mask's value is not a decimal 0 or 255.

- The magic number is calculated as the value of the IP address's interesting octet, subtracted from 256.
- The subnet number can be found by copying the IP address octets to the left of the interesting octet; writing down 0s for octets to the right of the interesting octet; and by finding the multiple of the magic number closest to, but not larger than, the IP address's value in that same octet.
- The broadcast address can be similarly found, by coping the subnet number's octets to the left of the interesting octet; writing 255s for octets to the right of the interesting octet; and by taking the subnet number's value in the interesting octet, adding the magic number, and subtracting 1.

Table D-7 shows the work for this problem, with some explanation of the work following the table. Please refer to the reference pages in Appendix E for the detailed processes.

Table D-7 *Question 1: Subnet, Broadcast, First and Last Addresses Calculated Using Subnet Chart*

	Octet 1	Octet 2	Octet 3	Octet 4	Comments
Mask	255	192	0	0	
Address	10	180	10	18	
Subnet Number	10	128	0	0	Magic number = 256 – 192 = 64
First Address	10	128	0	1	Add 1 to last octet of subnet
Last Address	10	191	255	254	Subtract 1 from last octet of broadcast
Broadcast	10	191	255	255	128 + 64 – 1 = 191

This subnetting scheme uses a difficult mask because one of the octets is not a 0 or a 255. The second octet is "interesting" in this case. The key part of the trick to get the right answers is to calculate the magic number, which is 256 – 192 = 64 in this case (256 – mask's value in the interesting octet). The subnet number's value in the interesting octet (inside the box) is the multiple of the magic number that is not higher than the original IP address's value in the interesting octet. In this case, 128 is the multiple of 64 that is closest to 180 but not higher than 180. So, the second octet of the subnet number is 128.

The second part of this process calculates the subnet broadcast address, with the tricky part, as usual, in the "interesting" octet. Take the subnet number's value in the interesting octet, add the magic number, and subtract 1. That is the broadcast address's value in the interesting octet. In this case, 128 + 64 – 1 = 191.

If the steps are not apparent when comparing Table D-7 to the process summary RP-5C and RP-6C in Appendix E, you may want to view the subnetting videos found with this book. Subnetting videos 1, 2, and 3 show three examples that follow the exact steps in RP-5C (to find a subnet number). Subnetting videos 4, 5, and 6 show how to follow process RP-6C to find the broadcast address and range of assignable addresses for the same address/mask used in videos 1, 2, and 3. The videos can much more easily show the movement and actions taken with these processes.

Answer to Problem 2 in Problem Set 5

Table D-8 *Question 2: Size of Network, Subnet, Host, Number of Subnets, Number of Hosts*

Item	Example	Rules to Remember
Address	10.200.10.18	—
Mask	255.224.0.0	—
Number of network bits	8	Always defined by Class A, B, C
Number of host bits	21	Always defined as number of binary 0s in mask
Number of subnet bits	3	32 – (network size + host size)
Number of subnets	$2^3 = 8$	$2^{\text{number-of-subnet-bits}}$
Number of hosts	$2^{21} - 2 = 2{,}097{,}150$	$2^{\text{number-of-host-bits}} - 2$

Table D-9 contains the important binary calculations for finding the subnet number and subnet broadcast address, as summarized in Appendix E, RP-5A. To calculate the subnet number, perform a Boolean AND on the address and mask. To find the broadcast address for this subnet, change all the host bits to binary 1s in the subnet number. The host bits are in **bold** print in the table.

Table D-9 *Question 2: Binary Calculation of Subnet and Broadcast Addresses*

Address	10.200.10.18	00001010 110**01000 00001010 00010010**
Mask	255.224.0.0	11111111 111**00000 00000000 00000000**
AND result (subnet number)	10.192.0.0	00001010 110**00000 00000000 00000000**
Change host to 1s (broadcast address)	10.223.255.255	00001010 110**11111 11111111 11111111**

Just add 1 to the subnet number to get the first valid IP address; just subtract 1 from the broadcast address to get the last valid IP address. In this case:

10.192.0.1 through 10.223.255.254

Alternately, you can use the processes that only use decimal math to find the subnet and broadcast address. Table D-10 shows the work for this problem, with some explanation of the work following the table.

Table D-10 *Question 2: Subnet, Broadcast, First and Last Addresses Calculated Using Subnet Chart*

	Octet 1	Octet 2	Octet 3	Octet 4	Comments
Mask	255	224	0	0	
Address	10	200	10	18	
Subnet Number	10	192	0	0	Magic number = 256 – 224 = 32
First Address	10	192	0	1	Add 1 to last octet of subnet
Last Address	10	223	255	254	Subtract 1 from last octet of broadcast
Broadcast	10	223	255	255	192 + 32 – 1 = 223

This subnetting scheme uses a difficult mask because one of the octets is not a 0 or a 255. The second octet is "interesting" in this case. The key part of the trick to get the right answers is to calculate the magic number, which is 256 – 224 = 32 in this case (256 – mask's value in the interesting octet). The subnet number's value in the interesting octet (inside the box) is the multiple of the magic number that is not higher than the original IP address's value in the interesting octet. In this case, 192 is the multiple of 32 that is closest to 200 but not higher than 200. So, the second octet of the subnet number is 192.

The second part of this process calculates the subnet broadcast address, with the tricky part, as usual, in the "interesting" octet. Take the subnet number's value in the interesting octet, add the magic number, and subtract 1. That is the broadcast address's value in the interesting octet. In this case, 192 + 32 – 1 = 223.

Answer to Problem 3 in Problem Set 5

Table D-11 *Question 3: Size of Network, Subnet, Host, Number of Subnets, Number of Hosts*

Item	Example	Rules to Remember
Address	10.100.18.18	—
Mask	255.240.0.0	—
Number of network bits	8	Always defined by Class A, B, C
Number of host bits	20	Always defined as number of binary 0s in mask
Number of subnet bits	4	32 – (network size + host size)
Number of subnets	$2^4 = 16$	$2^{\text{number-of-subnet-bits}}$
Number of hosts	$2^{20} - 2 = 1{,}048{,}574$	$2^{\text{number-of-host-bits}} - 2$

Table D-12 contains the important binary calculations for finding the subnet number and subnet broadcast address, as summarized in Appendix E, RP-5A. To calculate the subnet number, perform a Boolean AND on the address and mask. To find the broadcast address for this subnet, change all the host bits to binary 1s in the subnet number. The host bits are in **bold** print in the table.

Table D-12 *Question 3: Binary Calculation of Subnet and Broadcast Addresses*

Address	10.100.18.18	00001010 0110**0100 00010010 00010010**
Mask	255.240.0.0	11111111 1111**0000 00000000 00000000**
AND result (subnet number)	10.96.0.0	00001010 0110**0000 00000000 00000000**
Change host to 1s (broadcast address)	10.111.255.255	00001010 0110**1111 11111111 11111111**

Just add 1 to the subnet number to get the first valid IP address; just subtract 1 from the broadcast address to get the last valid IP address. In this case:

10.96.0.1 through 10.111.255.254

Alternately, you can use the processes that only use decimal math to find the subnet and broadcast address. Table D-13 shows the work for this problem, with some explanation of the work following the table.

Table D-13 *Question 3: Subnet, Broadcast, First and Last Addresses Calculated Using Subnet Chart*

	Octet 1	Octet 2	Octet 3	Octet 4	Comments
Mask	255	240	0	0	—
Address	10	100	18	18	—
Subnet Number	10	96	0	0	Magic number = 256 – 240 = 16
First Address	10	96	0	1	Add 1 to last octet of subnet
Last Address	10	111	255	254	Subtract 1 from last octet of broadcast
Broadcast	10	111	255	255	96 + 16 – 1 = 111

This subnetting scheme uses a difficult mask because one of the octets is not a 0 or a 255. The second octet is "interesting" in this case. The key part of the trick to get the right answers is to calculate the magic number, which is 256 – 240 = 16 in this case (256 – mask's value in the interesting octet). The subnet number's value in the interesting octet (inside the box) is the multiple of the magic number that is not higher than the original IP address's value in the interesting octet. In this case, 96 is the multiple of 16 that is closest to 100 but not higher than 100. So, the second octet of the subnet number is 96.

The second part of this process calculates the subnet broadcast address, with the tricky part, as usual, in the "interesting" octet. Take the subnet number's value in the interesting octet, add the magic number, and subtract 1. That is the broadcast address's value in the interesting octet. In this case, 96 + 16 – 1 = 111.

Answer to Problem 4 in Problem Set 5

Table D-14 *Question 4: Size of Network, Subnet, Host, Number of Subnets, Number of Hosts*

Item	Example	Rules to Remember
Address	10.100.18.18	—
Mask	255.248.0.0	—
Number of network bits	8	Always defined by Class A, B, C
Number of host bits	19	Always defined as number of binary 0s in mask
Number of subnet bits	5	32 – (network size + host size)
Number of subnets	$2^5 = 32$	$2^{\text{number-of-subnet-bits}}$
Number of hosts	$2^{19} - 2 = 524,286$	$2^{\text{number-of-host-bits}} - 2$

Table D-15 contains the important binary calculations for finding the subnet number and subnet broadcast address, as summarized in Appendix E, RP-5A. To calculate the subnet number, perform a Boolean AND on the address and mask. To find the broadcast address for this subnet, change all the host bits to binary 1s in the subnet number. The host bits are in **bold** print in the table.

Table D-15 *Question 4: Binary Calculation of Subnet and Broadcast Addresses*

Address	10.100.18.18	`00001010 01100`**`100 00010010 00010010`**
Mask	255.248.0.0	`11111111 11111`**`000 00000000 00000000`**
AND result (subnet number)	10.96.0.0	`00001010 01100`**`000 00000000 00000000`**
Change host to 1s (broadcast address)	10.103.255.255	`00001010 01100`**`111 11111111 11111111`**

Just add 1 to the subnet number to get the first valid IP address; just subtract 1 from the broadcast address to get the last valid IP address. In this case:

10.96.0.1 through 10.103.255.254

Alternately, you can use the processes that only use decimal math to find the subnet and broadcast address. Table D-16 shows the work for this problem, with some explanation of the work following the table.

Table D-16 *Question 4: Subnet, Broadcast, First and Last Addresses Calculated Using Subnet Chart*

	Octet 1	Octet 2	Octet 3	Octet 4	Comments
Mask	255	248	0	0	—
Address	10	100	18	18	—
Subnet Number	10	96	0	0	Magic number = 256 – 248 = 8
First Address	10	96	0	1	Add 1 to last octet of subnet
Last Address	10	103	255	254	Subtract 1 from last octet of broadcast
Broadcast	10	103	255	255	96 + 8 – 1 = 103

This subnetting scheme uses a difficult mask because one of the octets is not a 0 or a 255. The second octet is "interesting" in this case. The key part of the trick to get the right answers is to calculate the magic number, which is 256 – 248 = 8 in this case (256 – mask's value in the interesting octet). The subnet number's value in the interesting octet (inside the box) is the multiple of the magic number that is not higher than the original IP address's value in the interesting octet. In this case, 96 is the multiple of 8 that is closest to 100 but not higher than 100. So, the second octet of the subnet number is 96.

The second part of this process calculates the subnet broadcast address with the tricky part, as usual, in the "interesting" octet. Take the subnet number's value in the interesting octet, add the magic number, and subtract 1. That is the broadcast address's value in the interesting octet. In this case, 96 + 8 – 1 = 103.

Answer to Problem 5 in Problem Set 5

Table D-17 *Question 5: Size of Network, Subnet, Host, Number of Subnets, Number of Hosts*

Item	Example	Rules to Remember
Address	10.150.200.200	—
Mask	255.252.0.0	—
Number of network bits	8	Always defined by Class A, B, C
Number of host bits	18	Always defined as number of binary 0s in mask
Number of subnet bits	6	32 – (network size + host size)
Number of subnets	$2^6 = 64$	$2^{\text{number-of-subnet-bits}}$
Number of hosts	$2^{18} - 2 = 262{,}142$	$2^{\text{number-of-host-bits}} - 2$

Table D-18 contains the important binary calculations for finding the subnet number and subnet broadcast address, as summarized in Appendix E, RP-5A. To calculate the subnet number, perform a Boolean AND on the address and mask. To find the broadcast address for this subnet, change all the host bits to binary 1s in the subnet number. The host bits are in **bold** print in the table.

Table D-18 *Question 5: Binary Calculation of Subnet and Broadcast Addresses*

Address	10.150.200.200	`00001010 100101`**`10 11001000 11001000`**
Mask	255.252.0.0	`11111111 111111`**`00 00000000 00000000`**
AND result (subnet number)	10.148.0.0	`00001010 100101`**`00 00000000 00000000`**
Change host to 1s (broadcast address)	10.151.255.255	`00001010 100101`**`11 11111111 11111111`**

Just add 1 to the subnet number to get the first valid IP address; just subtract 1 from the broadcast address to get the last valid IP address. In this case:

10.148.0.1 through 10.151.255.254

Alternately, you can use the processes that only use decimal math to find the subnet and broadcast address. Table D-19 shows the work for this problem, with some explanation of the work following the table.

Table D-19 *Question 5: Subnet, Broadcast, First and Last Addresses Calculated Using Subnet Chart*

	Octet 1	Octet 2	Octet 3	Octet 4	Comments
Mask	255	252	0	0	—
Address	10	150	200	200	—
Subnet Number	10	148	0	0	Magic number = 256 – 252 = 4
First Address	10	148	0	1	Add 1 to last octet of subnet
Last Address	10	151	255	254	Subtract 1 from last octet of broadcast
Broadcast	10	151	255	255	148 + 4 – 1 = 151

This subnetting scheme uses a difficult mask because one of the octets is not a 0 or a 255. The second octet is "interesting" in this case. The key part of the trick to get the right answers is to calculate the magic number, which is 256 – 252 = 4 in this case (256 – mask's value in the interesting octet). The subnet number's value in the interesting octet (inside the box) is the multiple of the magic number that is not higher than the original IP address's value in the interesting octet. In this case, 148 is the multiple of 4 that is closest to 150 but not higher than 150. So, the second octet of the subnet number is 148.

The second part of this process calculates the subnet broadcast address with the tricky part, as usual, in the "interesting" octet. Take the subnet number's value in the interesting octet, add the magic number, and subtract 1. That is the broadcast address's value in the interesting octet. In this case, 148 + 4 – 1 = 151.

Answer to Problem 6 in Problem Set 5

Table D-20 *Question 6: Size of Network, Subnet, Host, Number of Subnets, Number of Hosts*

Item	Example	Rules to Remember
Address	10.150.200.200	—
Mask	255.254.0.0	—
Number of network bits	8	Always defined by Class A, B, C
Number of host bits	17	Always defined as number of binary 0s in mask
Number of subnet bits	7	32 – (network size + host size)
Number of subnets	$2^7 = 128$	$2^{\text{number-of-subnet-bits}}$
Number of hosts	$2^{17} - 2 = 131{,}070$	$2^{\text{number-of-host-bits}} - 2$

Table D-21 contains the important binary calculations for finding the subnet number and subnet broadcast address, as summarized in Appendix E, RP-5A. To calculate the subnet number, perform a Boolean AND on the address and mask. To find the broadcast address for this subnet, change all the host bits to binary 1s in the subnet number. The host bits are in **bold** print in the table.

Table D-21 *Question 6: Binary Calculation of Subnet and Broadcast Addresses*

Address	10.150.200.200	`00001010 1001011`**`0 11001000 11001000`**
Mask	255.254.0.0	`11111111 1111111`**`0 00000000 00000000`**
AND result (subnet number)	10.150.0.0	`00001010 1001011`**`0 00000000 00000000`**
Change host to 1s (broadcast address)	10.151.255.255	`00001010 1001011`**`1 11111111 11111111`**

Just add 1 to the subnet number to get the first valid IP address; just subtract 1 from the broadcast address to get the last valid IP address. In this case:

10.150.0.1 through 10.151.255.254

Alternately, you can use the processes that only use decimal math to find the subnet and broadcast address. Table D-22 shows the work for this problem, with some explanation of the work following the table.

Table D-22 *Question 6: Subnet, Broadcast, First and Last Addresses Calculated Using Subnet Chart*

	Octet 1	Octet 2	Octet 3	Octet 4
Mask	255	254	0	0
Address	10	150	200	200
Subnet Number	10	150	0	0
First Valid Address	10	150	0	1
Last Valid Address	10	151	255	254
Broadcast	10	151	255	255

This subnetting scheme uses a difficult mask because one of the octets is not a 0 or a 255. The second octet is "interesting" in this case. The key part of the trick to get the right answers is to calculate the magic number, which is 256 – 254 = 2 in this case (256 – mask's value in the interesting octet). The subnet number's value in the interesting octet (inside the box) is the multiple of the magic number that is not higher than the original IP address's value in the interesting octet. In this case, 150 is the multiple of 2 that is closest to 150 but not higher than 150. So, the second octet of the subnet number is 150.

The second part of this process calculates the subnet broadcast address with the tricky part, as usual, in the "interesting" octet. Take the subnet number's value in the interesting octet, add the magic number, and subtract 1. That is the broadcast address's value in the interesting octet. In this case, 150 + 2 – 1 = 151.

Answer to Problem 7 in Problem Set 5

Table D-23 *Question 7: Size of Network, Subnet, Host, Number of Subnets, Number of Hosts*

Item	Example	Rules to Remember
Address	10.220.100.18	—
Mask	255.255.0.0	—
Number of network bits	8	Always defined by Class A, B, C
Number of host bits	16	Always defined as number of binary 0s in mask
Number of subnet bits	8	32 – (network size + host size)
Number of subnets	$2^8 = 256$	$2^{\text{number-of-subnet-bits}}$
Number of hosts	$2^{16} - 2 = 65,534$	$2^{\text{number-of-host-bits}} - 2$

Table D-24 contains the important binary calculations for finding the subnet number and subnet broadcast address, as summarized in Appendix E, RP-5A. To calculate the subnet number, perform a Boolean AND on the address and mask. To find the broadcast address for this subnet, change all the host bits to binary 1s in the subnet number. The host bits are in **bold** print in the table.

Table D-24 *Question 7: Binary Calculation of Subnet and Broadcast Addresses*

Address	10.220.100.18	00001010 11011100 **01100100 00010010**
Mask	255.255.0.0	11111111 11111111 **00000000 00000000**
AND result (subnet number)	10.220.0.0	00001010 11011100 **00000000 00000000**
Change host to 1s (broadcast address)	10.220.255.255	00001010 11011100 **11111111 11111111**

Just add 1 to the subnet number to get the first valid IP address; just subtract 1 from the broadcast address to get the last valid IP address. In this case:

10.220.0.1 through 10.220.255.254

Alternately, you can use the processes that only use decimal math to find the subnet and broadcast address. Table D-25 shows the work for this problem, with some explanation of the work following the table.

Table D-25 *Question 7: Subnet, Broadcast, First, and Last Addresses Calculated Using Subnet Chart*

	Octet 1	Octet 2	Octet 3	Octet 4
Mask	255	255	0	0
Address	10	220	100	18
Subnet Number	10	220	0	0
First Valid Address	10	220	0	1
Last Valid Address	10	220	255	254
Broadcast	10	220	255	255

This subnetting scheme uses an easy mask because all of the octets are a 0 or a 255. No math tricks are needed at all.

Answer to Problem 8 in Problem Set 5

Table D-26 *Question 8: Size of Network, Subnet, Host, Number of Subnets, Number of Hosts*

Item	Example	Rules to Remember
Address	10.220.100.18	—
Mask	255.255.128.0	—
Number of network bits	8	Always defined by Class A, B, C
Number of host bits	15	Always defined as number of binary 0s in mask
Number of subnet bits	9	32 – (network size + host size)
Number of subnets	$2^9 = 510$	$2^{\text{number-of-subnet-bits}}$
Number of hosts	$2^{15} - 2 = 32{,}766$	$2^{\text{number-of-host-bits}} - 2$

Table D-27 contains the important binary calculations for finding the subnet number and subnet broadcast address, as summarized in Appendix E, RP-5A. To calculate the subnet number, perform a Boolean AND on the address and mask. To find the broadcast address for this subnet, change all the host bits to binary 1s in the subnet number. The host bits are in **bold** print in the table.

Table D-27 *Question 8: Binary Calculation of Subnet and Broadcast Addresses*

Address	10.220.100.18	`00001010 11011100 0`**`1100100 00010010`**
Mask	255.255.128.0	`11111111 11111111 1`**`0000000 00000000`**
AND result (subnet number)	10.220.0.0	`00001010 11011100 0`**`0000000 00000000`**
Change host to 1s (broadcast address)	10.220.127.255	`00001010 11011100 0`**`1111111 11111111`**

Just add 1 to the subnet number to get the first valid IP address; just subtract 1 from the broadcast address to get the last valid IP address. In this case:

10.220.0.1 through 10.220.127.254

Alternately, you can use the processes from Chapter 4 and from Appendix E (RP-5B, RP-5C, RP-6B, and RP-6C) that only use decimal math to find the subnet and broadcast address. Table D-28 shows the work for this problem, with some explanation of the work following the table. Please refer to Chapter 4 or to the reference pages in Appendix E for the detailed processes.

Table D-28 *Question 8: Subnet, Broadcast, First and Last Addresses Calculated Using Subnet Chart*

	Octet 1	Octet 2	Octet 3	Octet 4
Mask	255	255	128	0
Address	10	220	100	18
Subnet Number	10	220	0	0
First Address	10	220	0	1
Last Address	10	220	127	254
Broadcast	10	220	127	255

This subnetting scheme uses a difficult mask because one of the octets is not a 0 or a 255. The third octet is "interesting" in this case. The key part of the trick to get the right answers is to calculate the magic number, which is 256 – 128 = 128 in this case (256 – mask's value in the interesting octet). The subnet number's value in the interesting octet (inside the box) is the multiple of the magic number that is not higher than the original IP address's value in the interesting octet. In this case, 0 is the multiple of 128 that is closest to 100 but not higher than 100. So, the third octet of the subnet number is 0.

The second part of this process calculates the subnet broadcast address with the tricky part, as usual, in the "interesting" octet. Take the subnet number's value in the interesting octet, add the magic number, and subtract 1. That is the broadcast address's value in the interesting octet. In this case, 0 + 128 – 1 = 127.

This example tends to confuse people because a mask with 128 in it gives you subnet numbers that just do not seem to look right. Table D-29 gives you the answers for the first several subnets, just to make sure that you are clear about the subnets when using this mask with a Class A network.

Table D-29 *Question 8: First 4 Subnets*

	Zero Subnet	2nd Subnet	3rd Subnet	4th Subnet
Subnet	10.0.0.0	10.0.128.0	10.1.0.0	10.1.128.0
First Address	10.0.0.1	10.0.128.1	10.1.0.1	10.1.128.1
Last Address	10.0.127.254	10.0.255.254	10.1.127.254	10.1.255.254
Broadcast	10.0.127.255	10.0.255.255	10.1.127.255	10.1.255.255

Answer to Problem 9 in Problem Set 5

Table D-30 *Question 9: Size of Network, Subnet, Host, Number of Subnets, Number of Hosts*

Item	Example	Rules to Remember
Address	172.31.100.100	—
Mask	255.255.192.0	—
Number of network bits	16	Always defined by Class A, B, C
Number of host bits	14	Always defined as number of binary 0s in mask
Number of subnet bits	2	32 – (network size + host size)
Number of subnets	$2^2 = 4$	$2^{\text{number-of-subnet-bits}}$
Number of hosts	$2^{14} - 2 = 16{,}382$	$2^{\text{number-of-host-bits}} - 2$

Table D-31 contains the important binary calculations for finding the subnet number and subnet broadcast address, as summarized in Appendix E, RP-5A. To calculate the subnet number, perform a Boolean AND on the address and mask. To find the broadcast address for this subnet, change all the host bits to binary 1s in the subnet number. The host bits are in **bold** print in the table.

Table D-31 *Question 9: Binary Calculation of Subnet and Broadcast Addresses*

Address	172.31.100.100	10101100 00011111 01**100100 01100100**
Mask	255.255.192.0	11111111 11111111 11**000000 00000000**
AND result (subnet number)	172.31.64.0	10101100 00011111 01**000000 00000000**
Change host to 1s (broadcast address)	172.31.127.255	10101100 00011111 01**111111 11111111**

Just add 1 to the subnet number to get the first valid IP address; just subtract 1 from the broadcast address to get the last valid IP address. In this case:

172.31.64.1 through 172.31.127.254

Alternately, you can use the processes that only use decimal math to find the subnet and broadcast address. Table D-32 shows the work for this problem, with some explanation of the work following the table.

Table D-32 *Question 9: Subnet, Broadcast, First and Last Addresses Calculated Using Subnet Chart*

	Octet 1	Octet 2	Octet 3	Octet 4
Mask	255	255	192	0
Address	172	31	100	100
Subnet Number	172	31	64	0
First Valid Address	172	31	64	1
Last Valid Address	172	31	127	254
Broadcast	172	31	127	255

This subnetting scheme uses a difficult mask because one of the octets is not a 0 or a 255. The third octet is "interesting" in this case. The key part of the trick to get the right answers is to calculate the magic number, which is 256 – 192 = 64 in this case (256 – mask's value in the interesting octet). The subnet number's value in the interesting octet (inside the box) is the multiple of the magic number that is not higher than the original IP address's value in the interesting octet. In this case, 64 is the multiple of 64 that is closest to 100 but not higher than 100. So, the third octet of the subnet number is 64.

The second part of this process calculates the subnet broadcast address with the tricky part, as usual, in the "interesting" octet. Take the subnet number's value in the interesting octet, add the magic number, and subtract 1. That is the broadcast address's value in the interesting octet. In this case, 64 + 64 – 1 = 127.

Answer to Problem 10 in Problem Set 5

Table D-33 *Question 10: Size of Network, Subnet, Host, Number of Subnets, Number of Hosts*

Item	Example	Rules to Remember
Address	172.31.100.100	—
Mask	255.255.224.0	—
Number of network bits	16	Always defined by Class A, B, C
Number of host bits	13	Always defined as number of binary 0s in mask
Number of subnet bits	3	32 – (network size + host size)
Number of subnets	$2^3 = 8$	$2^{\text{number-of-subnet-bits}}$
Number of hosts	$2^{13} - 2 = 8190$	$2^{\text{number-of-host-bits}} - 2$

Table D-34 contains the important binary calculations for finding the subnet number and subnet broadcast address, as summarized in Appendix E, RP-5A. To calculate the subnet number, perform a Boolean AND on the address and mask. To find the broadcast address for this subnet, change all the host bits to binary 1s in the subnet number. The host bits are in **bold** print in the table.

Table D-34 *Question 10: Binary Calculation of Subnet and Broadcast Addresses*

Address	172.31.100.100	10101100 00011111 011**00100 01100100**
Mask	255.255.224.0	11111111 11111111 111**00000 00000000**
AND result (subnet number)	172.31.96.0	10101100 00011111 011**00000 00000000**
Change host to 1s (broadcast address)	172.31.127.255	10101100 00011111 011**11111 11111111**

Just add 1 to the subnet number to get the first valid IP address; just subtract 1 from the broadcast address to get the last valid IP address. In this case:

172.31.96.1 through 172.31.127.254

Alternately, you can use the processes that only use decimal math to find the subnet and broadcast address. Table D-35 shows the work for this problem, with some explanation of the work following the table.

Table D-35 *Question 10: Subnet, Broadcast, First and Last Addresses Calculated Using Subnet Chart*

	Octet 1	Octet 2	Octet 3	Octet 4
Mask	255	255	224	0
Address	172	31	100	100
Subnet Number	172	31	96	0
First Valid Address	172	31	96	1
Last Valid Address	172	31	127	254
Broadcast	172	31	127	255

This subnetting scheme uses a difficult mask because one of the octets is not a 0 or a 255. The third octet is "interesting" in this case. The key part of the trick to get the right answers is to calculate the magic number, which is 256 – 224 = 32 in this case (256 – mask's value in the interesting octet). The subnet number's value in the interesting octet (inside the box) is the multiple of the magic number that is not higher than the original IP address's value in the interesting octet. In this case, 96 is the multiple of 32 that is closest to 100 but not higher than 100. So, the third octet of the subnet number is 96.

The second part of this process calculates the subnet broadcast address, with the tricky parts, as usual, in the "interesting" octet. Take the subnet number's value in the interesting octet, add the magic number, and subtract 1. That is the broadcast address's value in the interesting octet. In this case, 96 + 32 – 1 = 127.

Answer to Problem 11 in Problem Set 5

Table D-36 *Question 11: Size of Network, Subnet, Host, Number of Subnets, Number of Hosts*

Item	Example	Rules to Remember
Address	172.31.200.10	—
Mask	255.255.240.0	—
Number of network bits	16	Always defined by Class A, B, C
Number of host bits	12	Always defined as number of binary 0s in mask
Number of subnet bits	4	32 – (network size + host size)
Number of subnets	$2^4 = 16$	$2^{\text{number-of-subnet-bits}}$
Number of hosts	$2^{12} - 2 = 4094$	$2^{\text{number-of-host-bits}} - 2$

Table D-37 contains the important binary calculations for finding the subnet number and subnet broadcast address, as summarized in Appendix E, RP-5A. To calculate the subnet number, perform a Boolean AND on the address and mask. To find the broadcast address for this subnet, change all the host bits to binary 1s in the subnet number. The host bits are in **bold** print in the table.

Table D-37 *Question 11: Binary Calculation of Subnet and Broadcast Addresses*

Address	172.31.200.10	`10101100 00011111 1100`**`1000 00001010`**
Mask	255.255.240.0	`11111111 11111111 1111`**`0000 00000000`**
AND result (subnet number)	172.31.192.0	`10101100 00011111 1100`**`0000 00000000`**
Change host to 1s (broadcast address)	172.31.207.255	`10101100 00011111 1100`**`1111 11111111`**

Just add 1 to the subnet number to get the first valid IP address; just subtract 1 from the broadcast address to get the last valid IP address. In this case:

172.31.192.1 through 172.31.207.254

Alternately, you can use the processes that only use decimal math to find the subnet and broadcast address. Table D-38 shows the work for this problem, with some explanation of the work following the table.

Table D-38 *Question 11: Subnet, Broadcast, First and Last Addresses Calculated Using Subnet Chart*

	Octet 1	Octet 2	Octet 3	Octet 4
Mask	255	255	240	0
Address	172	31	200	10
Subnet Number	172	31	192	0
First Valid Address	172	31	192	1
Last Valid Address	172	31	207	254
Broadcast	172	31	207	255

This subnetting scheme uses a difficult mask because one of the octets is not a 0 or a 255. The third octet is "interesting" in this case. The key part of the trick to get the right answers is to calculate the magic number, which is 256 – 240 = 16 in this case (256 – mask's value in the interesting octet). The subnet number's value in the interesting octet (inside the box) is the multiple of the magic number that is not higher than the original IP address's value in the interesting octet. In this case, 192 is the multiple of 16 that is closest to 200 but not higher than 200. So, the third octet of the subnet number is 192.

The second part of this process calculates the subnet broadcast address with the tricky part, as usual, in the "interesting" octet. Take the subnet number's value in the interesting octet, add the magic number, and subtract 1. That is the broadcast address's value in the interesting octet. In this case, 192 + 16 – 1 = 207.

Answer to Problem 12 in Problem Set 5

Table D-39 *Question 12: Size of Network, Subnet, Host, Number of Subnets, Number of Hosts*

Item	Example	Rules to Remember
Address	172.31.200.10	—
Mask	255.255.248.0	—
Number of network bits	16	Always defined by Class A, B, C
Number of host bits	11	Always defined as number of binary 0s in mask
Number of subnet bits	5	32 – (network size + host size)
Number of subnets	$2^5 = 32$	$2^{\text{number-of-subnet-bits}}$
Number of hosts	$2^{11} - 2 = 2046$	$2^{\text{number-of-host-bits}} - 2$

Table D-40 contains the important binary calculations for finding the subnet number and subnet broadcast address, as summarized in Appendix E, RP-5A. To calculate the subnet number, perform a Boolean AND on the address and mask. To find the broadcast address for this subnet, change all the host bits to binary 1s in the subnet number. The host bits are in **bold** print in the table.

Table D-40 *Question 12: Binary Calculation of Subnet and Broadcast Addresses*

Address	172.31.200.10	`10101100 00011111 11001`**`000 00001010`**
Mask	255.255.248.0	`11111111 11111111 11111`**`000 00000000`**
AND result (subnet number)	172.31.200.0	`10101100 00011111 11001`**`000 00000000`**
Change host to 1s (broadcast address)	172.31.207.255	`10101100 00011111 11001`**`111 11111111`**

Just add 1 to the subnet number to get the first valid IP address; just subtract 1 from the broadcast address to get the last valid IP address. In this case:

172.31.200.1 through 172.31.207.254

Alternately, you can use the processes that only use decimal math to find the subnet and broadcast address. Table D-41 shows the work for this problem, with some explanation of the work following the table.

Table D-41 *Question 12: Subnet, Broadcast, First and Last Addresses Calculated Using Subnet Chart*

	Octet 1	Octet 2	Octet 3	Octet 4
Mask	255	255	248	0
Address	172	31	200	10
Subnet Number	172	31	200	0
First Valid Address	172	31	200	1
Last Valid Address	172	31	207	254
Broadcast	172	31	207	255

This subnetting scheme uses a difficult mask because one of the octets is not a 0 or a 255. The third octet is "interesting" in this case. The key part of the trick to get the right answers is to calculate the magic number, which is 256 – 248 = 8 in this case (256 – mask's value in the interesting octet). The subnet number's value in the interesting octet (inside the box) is the multiple of the magic number that is not higher than the original IP address's value in the interesting octet. In this case, 200 is the multiple of 8 that is closest to 200 but not higher than 200. So, the third octet of the subnet number is 200.

The second part of this process calculates the subnet broadcast address, with the tricky part, as usual, in the "interesting" octet. Take the subnet number's value in the interesting octet, add the magic number, and subtract 1. That is the broadcast address's value in the interesting octet. In this case, 200 + 8 – 1 = 207.

Answer to Problem 13 in Problem Set 5

Table D-42 *Question 13: Size of Network, Subnet, Host, Number of Subnets, Number of Hosts*

Item	Example	Rules to Remember
Address	172.31.50.50	—
Mask	255.255.252.0	—
Number of network bits	16	Always defined by Class A, B, C
Number of host bits	10	Always defined as number of binary 0s in mask
Number of subnet bits	6	32 – (network size + host size)
Number of subnets	$2^6 = 64$	$2^{\text{number-of-subnet-bits}}$
Number of hosts	$2^{10} - 2 = 1022$	$2^{\text{number-of-host-bits}} - 2$

Table D-43 contains the important binary calculations for finding the subnet number and subnet broadcast address, as summarized in Appendix E, RP-5A. To calculate the subnet number, perform a Boolean AND on the address and mask. To find the broadcast address for this subnet, change all the host bits to binary 1s in the subnet number. The host bits are in **bold** print in the table.

Table D-43 *Question 13: Binary Calculation of Subnet and Broadcast Addresses*

Address	172.31.50.50	`10101100 00011111 001100`**`10 00110010`**
Mask	255.255.252.0	`11111111 11111111 111111`**`00 00000000`**
AND result (subnet number)	172.31.48.0	`10101100 00011111 001100`**`00 00000000`**
Change host to 1s (broadcast address)	172.31.51.255	`10101100 00011111 001100`**`11 11111111`**

Just add 1 to the subnet number to get the first valid IP address; just subtract 1 from the broadcast address to get the last valid IP address. In this case:

172.31.48.1 through 172.31.51.254

Alternately, you can use the processes that only use decimal math to find the subnet and broadcast address. Table D-44 shows the work for this problem, with some explanation of the work following the table.

Table D-44 *Question 13: Subnet, Broadcast, First and Last Addresses Calculated Using Subnet Chart*

	Octet 1	Octet 2	Octet 3	Octet 4
Mask	255	255	252	0
Address	172	31	50	50
Subnet Number	172	31	48	0
First Valid Address	172	31	48	1
Last Valid Address	172	31	51	254
Broadcast	172	31	51	255

This subnetting scheme uses a difficult mask because one of the octets is not a 0 or a 255. The third octet is "interesting" in this case. The key part of the trick to get the right answers is to calculate the magic number, which is 256 – 252 = 4 in this case (256 – mask's value in the interesting octet). The subnet number's value in the interesting octet (inside the box) is the multiple of the magic number that is not higher than the original IP address's value in the interesting octet. In this case, 48 is the multiple of 4 that is closest to 50 but not higher than 50. So, the third octet of the subnet number is 48.

The second part of this process calculates the subnet broadcast address, with the tricky part, as usual, in the "interesting" octet. Take the subnet number's value in the interesting octet, add the magic number, and subtract 1. That is the broadcast address's value in the interesting octet. In this case, 48 + 4 – 1 = 51.

Answer to Problem 14 in Problem Set 5

Table D-45 *Question 14: Size of Network, Subnet, Host, Number of Subnets, Number of Hosts*

Item	Example	Rules to Remember
Address	172.31.50.50	—
Mask	255.255.254.0	—
Number of network bits	16	Always defined by Class A, B, C
Number of host bits	9	Always defined as number of binary 0s in mask
Number of subnet bits	7	32 – (network size + host size)
Number of subnets	$2^7 = 128$	$2^{\text{number-of-subnet-bits}}$
Number of hosts	$2^9 - 2 = 510$	$2^{\text{number-of-host-bits}} - 2$

Table D-46 contains the important binary calculations for finding the subnet number and subnet broadcast address, as summarized in Appendix E, RP-5A. To calculate the subnet number, perform a Boolean AND on the address and mask. To find the broadcast address for this subnet, change all the host bits to binary 1s in the subnet number. The host bits are in **bold** print in the table.

Table D-46 *Question 14: Binary Calculation of Subnet and Broadcast Addresses*

Address	172.31.50.50	`10101100 00011111 0011001`**`0 00110010`**
Mask	255.255.254.0	`11111111 11111111 1111111`**`0 00000000`**
AND result (subnet number)	172.31.50.0	`10101100 00011111 0011001`**`0 00000000`**
Change host to 1s (broadcast address)	172.31.51.255	`10101100 00011111 0011001`**`1 11111111`**

Just add 1 to the subnet number to get the first valid IP address; just subtract 1 from the broadcast address to get the last valid IP address. In this case:

172.31.50.1 through 172.31.51.254

Alternately, you can use the processes that only use decimal math to find the subnet and broadcast address. Table D-47 shows the work for this problem, with some explanation of the work following the table.

Table D-47 *Question 14: Subnet, Broadcast, First and Last Addresses Calculated Using Subnet Chart*

	Octet 1	Octet 2	Octet 3	Octet 4
Mask	255	255	254	0
Address	172	31	50	50
Subnet Number	172	31	50	0
First Valid Address	172	31	50	1
Last Valid Address	172	31	51	254
Broadcast	172	31	51	255

This subnetting scheme uses a difficult mask because one of the octets is not a 0 or a 255. The third octet is “interesting” in this case. The key part of the trick to get the right answers is to calculate the magic number, which is 256 – 254 = 2 in this case (256 – mask’s value in the interesting octet). The subnet number’s value in the interesting octet (inside the box) is the multiple of the magic number that is not higher than the original IP address’s value in the interesting octet. In this case, 50 is the multiple of 2 that is closest to 50 but not higher than 50. So, the third octet of the subnet number is 50.

The second part of this process calculates the subnet broadcast address with the tricky part, as usual, in the “interesting” octet. Take the subnet number’s value in the interesting octet, add the magic number, and subtract 1. That is the broadcast address’s value in the interesting octet. In this case, 50 + 2 – 1 = 51.

Answer to Problem 15 in Problem Set 5

Table D-48 *Question 15: Size of Network, Subnet, Host, Number of Subnets, Number of Hosts*

Item	Example	Rules to Remember
Address	172.31.140.14	—
Mask	255.255.255.0	—
Number of network bits	16	Always defined by Class A, B, C
Number of host bits	8	Always defined as number of binary 0s in mask
Number of subnet bits	8	32 – (network size + host size)
Number of subnets	$2^8 = 256$	$2^{\text{number-of-subnet-bits}}$
Number of hosts	$2^8 - 2 = 254$	$2^{\text{number-of-host-bits}} - 2$

Table D-49 contains the important binary calculations for finding the subnet number and subnet broadcast address, as summarized in Appendix E, RP-5A. To calculate the subnet number, perform a Boolean AND on the address and mask. To find the broadcast address for this subnet, change all the host bits to binary 1s in the subnet number. The host bits are in **bold** print in the table.

Table D-49 *Question 15: Binary Calculation of Subnet and Broadcast Addresses*

Address	172.31.140.14	`10101100 00011111 10001100` **`00001110`**
Mask	255.255.255.0	`11111111 11111111 11111111` **`00000000`**
AND result (subnet number)	172.31.140.0	`10101100 00011111 10001100` **`00000000`**
Change host to 1s (broadcast address)	172.31.140.255	`10101100 00011111 10001100` **`11111111`**

Just add 1 to the subnet number to get the first valid IP address; just subtract 1 from the broadcast address to get the last valid IP address. In this case:

172.31.140.1 through 172.31.140.254

Alternately, you can use the processes that only use decimal math to find the subnet and broadcast address. Table D-50 shows the work for this problem, with some explanation of the work following the table.

Table D-50 *Question 15: Subnet, Broadcast, First and Last Addresses Calculated Using Subnet Chart*

	Octet 1	Octet 2	Octet 3	Octet 4
Mask	255	255	255	0
Address	172	31	140	14
Subnet Number	172	31	140	0
First Valid Address	172	31	140	1
Last Valid Address	172	31	140	254
Broadcast	172	31	140	255

This subnetting scheme uses an easy mask because all of the octets are a 0 or a 255. No math tricks are needed at all.

Answer to Problem 16 in Problem Set 5

Table D-51 *Question 16: Size of Network, Subnet, Host, Number of Subnets, Number of Hosts*

Item	Example	Rules to Remember
Address	172.31.140.14	—
Mask	255.255.255.128	—
Number of network bits	16	Always defined by Class A, B, C
Number of host bits	7	Always defined as number of binary 0s in mask
Number of subnet bits	9	32 – (network size + host size)
Number of subnets	$2^9 = 512$	$2^{\text{number-of-subnet-bits}}$
Number of hosts	$2^7 - 2 = 126$	$2^{\text{number-of-host-bits}} - 2$

Table D-52 contains the important binary calculations for finding the subnet number and subnet broadcast address, as summarized in Appendix E, RP-5A. To calculate the subnet number, perform a Boolean AND on the address and mask. To find the broadcast address for this subnet, change all the host bits to binary 1s in the subnet number. The host bits are in **bold** print in the table.

Table D-52 *Question 16: Binary Calculation of Subnet and Broadcast Addresses*

Address	172.31.140.14	10101100 00011111 10001100 0**0001110**
Mask	255.255.255.128	11111111 11111111 11111111 1**0000000**
AND result (subnet number)	172.31.140.0	10101100 00011111 10001100 0**0000000**
Change host to 1s (broadcast address)	172.31.140.127	10101100 00011111 10001100 0**1111111**

Just add 1 to the subnet number to get the first valid IP address; just subtract 1 from the broadcast address to get the last valid IP address. In this case:

172.31.140.1 through 172.31.140.126

Alternately, you can use the processes that only use decimal math to find the subnet and broadcast address. Table D-53 shows the work for this problem, with some explanation of the work following the table.

Table D-53 *Question 16: Subnet, Broadcast, First and Last Addresses Calculated Using Subnet Chart*

	Octet 1	Octet 2	Octet 3	Octet 4
Mask	255	255	255	128
Address	172	31	140	14
Subnet Number	172	31	140	0
First Valid Address	172	31	140	1
Last Valid Address	172	31	140	126
Broadcast	172	31	140	127

This subnetting scheme uses a difficult mask because one of the octets is not a 0 or a 255. The fourth octet is "interesting" in this case. The key part of the trick to get the right answers is to calculate the magic number, which is 256 – 128 = 128 in this case (256 – mask's value in the interesting octet). The subnet number's value in the interesting octet (inside the box) is the multiple of the magic number that is not higher than the original IP address's value in the interesting octet. In this case, 0 is the multiple of 128 that is closest to 14 but not higher than 14. So, the fourth octet of the subnet number is 0.

The second part of this process calculates the subnet broadcast address, with the tricky part, as usual, in the "interesting" octet. Take the subnet number's value in the interesting octet, add the magic number, and subtract 1. That is the broadcast address's value in the interesting octet. In this case, 0 + 128 – 1 = 127.

Answer to Problem 17 in Problem Set 5

Table D-54 *Question 17: Size of Network, Subnet, Host, Number of Subnets, Number of Hosts*

Item	Example	Rules to Remember
Address	192.168.15.150	—
Mask	255.255.255.192	—
Number of network bits	24	Always defined by Class A, B, C
Number of host bits	6	Always defined as number of binary 0s in mask
Number of subnet bits	2	32 – (network size + host size)
Number of subnets	$2^2 = 4$	$2^{\text{number-of-subnet-bits}}$
Number of hosts	$2^6 - 2 = 62$	$2^{\text{number-of-host-bits}} - 2$

Table D-55 contains the important binary calculations for finding the subnet number and subnet broadcast address, as summarized in Appendix E, RP-5A. To calculate the subnet number, perform a Boolean AND on the address and mask. To find the broadcast address for this subnet, change all the host bits to binary 1s in the subnet number. The host bits are in **bold** print in the table.

Table D-55 *Question 17: Binary Calculation of Subnet and Broadcast Addresses*

Address	192.168.15.150	11000000 10101000 00001111 10**010110**
Mask	255.255.255.192	11111111 11111111 11111111 11**000000**
AND result (subnet number)	192.168.15.128	11000000 10101000 00001111 10**000000**
Change host to 1s (broadcast address)	192.168.15.191	11000000 10101000 00001111 10**111111**

Just add 1 to the subnet number to get the first valid IP address; just subtract 1 from the broadcast address to get the last valid IP address. In this case:

192.168.15.129 through 192.168.15.190

Alternately, you can use the processes that only use decimal math to find the subnet and broadcast address. Table D-56 shows the work for this problem, with some explanation of the work following the table.

Table D-56 *Question 17: Subnet, Broadcast, First and Last Addresses Calculated Using Subnet Chart*

	Octet 1	Octet 2	Octet 3	Octet 4
Mask	255	255	255	192
Address	192	168	15	150
Subnet Number	192	168	15	128
First Valid Address	192	168	15	129
Last Valid Address	192	168	15	190
Broadcast	192	168	15	191

This subnetting scheme uses a difficult mask because one of the octets is not a 0 or a 255. The fourth octet is "interesting" in this case. The key part of the trick to get the right answers is to calculate the magic number, which is 256 – 192 = 64 in this case (256 – mask's value in the interesting octet). The subnet number's value in the interesting octet (inside the box) is the multiple of the magic number that is not higher than the original IP address's value in the interesting octet. In this case, 128 is the multiple of 64 that is closest to 150 but not higher than 150. So, the fourth octet of the subnet number is 128.

The second part of this process calculates the subnet broadcast address, with the tricky part, as usual, in the "interesting" octet. Take the subnet number's value in the interesting octet, add the magic number, and subtract 1. That is the broadcast address's value in the interesting octet. In this case, 128 + 64 – 1 = 191.

Answer to Problem 18 in Problem Set 5

Table D-57 *Question 18: Size of Network, Subnet, Host, Number of Subnets, Number of Hosts*

Item	Example	Rules to Remember
Address	192.168.15.150	—
Mask	255.255.255.224	—
Number of network bits	24	Always defined by Class A, B, C
Number of host bits	5	Always defined as number of binary 0s in mask
Number of subnet bits	3	32 – (network size + host size)
Number of subnets	$2^3 = 8$	$2^{\text{number-of-subnet-bits}}$
Number of hosts	$2^5 - 2 = 30$	$2^{\text{number-of-host-bits}} - 2$

Table D-58 contains the important binary calculations for finding the subnet number and subnet broadcast address, as summarized in Appendix E, RP-5A. To calculate the subnet number, perform a Boolean AND on the address and mask. To find the broadcast address for this subnet, change all the host bits to binary 1s in the subnet number. The host bits are in **bold** print in the table.

Table D-58 *Question 18: Binary Calculation of Subnet and Broadcast Addresses*

Address	192.168.15.150	`11000000 10101000 00001111 100`**`10110`**
Mask	255.255.255.224	`11111111 11111111 11111111 111`**`00000`**
AND result (subnet number)	192.168.15.128	`11000000 10101000 00001111 100`**`00000`**
Change host to 1s (broadcast address)	192.168.15.159	`11000000 10101000 00001111 100`**`11111`**

Just add 1 to the subnet number to get the first valid IP address; just subtract 1 from the broadcast address to get the last valid IP address. In this case:

192.168.15.129 through 192.168.15.158

Alternately, you can use the processes that only use decimal math to find the subnet and broadcast address. Table D-59 shows the work for this problem, with some explanation of the work following the table.

Table D-59 *Question 18: Subnet, Broadcast, First and Last Addresses Calculated Using Subnet Chart*

	Octet 1	Octet 2	Octet 3	Octet 4
Mask	255	255	255	224
Address	192	168	15	150
Subnet Number	192	168	15	128
First Valid Address	192	168	15	129
Last Valid Address	192	168	15	158
Broadcast	192	168	15	159

This subnetting scheme uses a difficult mask because one of the octets is not a 0 or a 255. The fourth octet is "interesting" in this case. The key part of the trick to get the right answers is to calculate the magic number, which is 256 – 224 = 32 in this case (256 – mask's value in the interesting octet). The subnet number's value in the interesting octet (inside the box) is the multiple of the magic number that is not higher than the original IP address's value in the interesting octet. In this case, 128 is the multiple of 32 that is closest to 150 but not higher than 150. So, the fourth octet of the subnet number is 128.

The second part of this process calculates the subnet broadcast address, with the tricky part, as usual, in the "interesting" octet. Take the subnet number's value in the interesting octet, add the magic number, and subtract 1. That is the broadcast address's value in the interesting octet. In this case, 128 + 32 – 1 = 159.

Answer to Problem 19 in Problem Set 5

Table D-60 *Question 19: Size of Network, Subnet, Host, Number of Subnets, Number of Hosts*

Item	Example	Rules to Remember
Address	192.168.100.100	—
Mask	255.255.255.240	—
Number of network bits	24	Always defined by Class A, B, C
Number of host bits	4	Always defined as number of binary 0s in mask
Number of subnet bits	4	32 – (network size + host size)
Number of subnets	$2^4 = 16$	$2^{\text{number-of-subnet-bits}}$
Number of hosts	$2^4 - 2 = 14$	$2^{\text{number-of-host-bits}} - 2$

Table D-61 contains the important binary calculations for finding the subnet number and subnet broadcast address, as summarized in Appendix E, RP-5A. To calculate the subnet number, perform a Boolean AND on the address and mask. To find the broadcast address for this subnet, change all the host bits to binary 1s in the subnet number. The host bits are in **bold** print in the table.

Table D-61 *Question 19: Binary Calculation of Subnet and Broadcast Addresses*

Address	192.168.100.100	`11000000 10101000 01100100 0110`**`0100`**
Mask	255.255.255.240	`11111111 11111111 11111111 1111`**`0000`**
AND result (subnet number)	192.168.100.96	`11000000 10101000 01100100 0110`**`0000`**
Change host to 1s (broadcast address)	192.168.100.111	`11000000 10101000 01100100 0110`**`1111`**

Just add 1 to the subnet number to get the first valid IP address; just subtract 1 from the broadcast address to get the last valid IP address. In this case:

192.168.100.97 through 192.168.100.110

Alternately, you can use the processes that only use decimal math to find the subnet and broadcast address. Table D-62 shows the work for this problem, with some explanation of the work following the table.

Table D-62 *Question 19: Subnet, Broadcast, First and Last Addresses Calculated Using Subnet Chart*

	Octet 1	Octet 2	Octet 3	Octet 4
Mask	255	255	255	240
Address	192	168	100	100
Subnet Number	192	168	100	96
First Valid Address	192	168	100	97
Last Valid Address	192	168	100	110
Broadcast	192	168	100	111

This subnetting scheme uses a difficult mask because one of the octets is not a 0 or a 255. The fourth octet is "interesting" in this case. The key part of the trick to get the right answers is to calculate the magic number, which is 256 – 240 = 16 in this case (256 – mask's value in the interesting octet). The subnet number's value in the interesting octet (inside the box) is the multiple of the magic number that is not higher than the original IP address's value in the interesting octet. In this case, 96 is the multiple of 16 that is closest to 100 but not higher than 100. So, the fourth octet of the subnet number is 96.

The second part of this process calculates the subnet broadcast address, with the tricky part, as usual, in the "interesting" octet. Take the subnet number's value in the interesting octet, add the magic number, and subtract 1. That is the broadcast address's value in the interesting octet. In this case, 96 + 16 – 1 = 111.

Answer to Problem 20 in Problem Set 5

Table D-63 *Question 20: Size of Network, Subnet, Host, Number of Subnets, Number of Hosts*

Item	Example	Rules to Remember
Address	192.168.100.100	—
Mask	255.255.255.248	—
Number of network bits	24	Always defined by Class A, B, C
Number of host bits	3	Always defined as number of binary 0s in mask
Number of subnet bits	5	32 – (network size + host size)
Number of subnets	$2^5 = 32$	$2^{\text{number-of-subnet-bits}}$
Number of hosts	$2^3 - 2 = 6$	$2^{\text{number-of-host-bits}} - 2$

Table D-64 contains the important binary calculations for finding the subnet number and subnet broadcast address, as summarized in Appendix E, RP-5A. To calculate the subnet number, perform a Boolean AND on the address and mask. To find the broadcast address for this subnet, change all the host bits to binary 1s in the subnet number. The host bits are in **bold** print in the table.

Table D-64 *Question 20: Binary Calculation of Subnet and Broadcast Addresses*

Address	192.168.100.100	`11000000 10101000 01100100 01100`**`100`**
Mask	255.255.255.248	`11111111 11111111 11111111 11111`**`000`**
AND result (subnet number)	192.168.100.96	`11000000 10101000 01100100 01100`**`000`**
Change host to 1s (broadcast address)	192.168.100.103	`11000000 10101000 01100100 01100`**`111`**

Just add 1 to the subnet number to get the first valid IP address; just subtract 1 from the broadcast address to get the last valid IP address. In this case:

192.168.100.97 through 192.168.100.102

Alternately, you can use the processes that only use decimal math to find the subnet and broadcast address. Table D-65 shows the work for this problem, with some explanation of the work following the table.

Table D-65 *Question 20: Subnet, Broadcast, First and Last Addresses Calculated Using Subnet Chart*

	Octet 1	Octet 2	Octet 3	Octet 4
Mask	255	255	255	248
Address	192	168	100	100
Subnet Number	192	168	100	96
First Valid Address	192	168	100	97
Last Valid Address	192	168	100	102
Broadcast	192	168	100	103

This subnetting scheme uses a difficult mask because one of the octets is not a 0 or a 255. The fourth octet is "interesting" in this case. The key part of the trick to get the right answers is to calculate the magic number, which is 256 – 248 = 8 in this case (256 – mask's value in the interesting octet). The subnet number's value in the interesting octet (inside the box) is the multiple of the magic number that is not higher than the original IP address's value in the interesting octet. In this case, 96 is the multiple of 8 that is closest to 100 but not higher than 100. So, the fourth octet of the subnet number is 96.

The second part of this process calculates the subnet broadcast address, with the tricky part, as usual, in the "interesting" octet. Take the subnet number's value in the interesting octet, add the magic number, and subtract 1. That is the broadcast address's value in the interesting octet. In this case, 96 + 8 – 1 = 103.

Answer to Problem 21 in Problem Set 5

Table D-66 *Question 21: Size of Network, Subnet, Host, Number of Subnets, Number of Hosts*

Item	Example	Rules to Remember
Address	192.168.15.230	—
Mask	255.255.255.252	—
Number of network bits	24	Always defined by Class A, B, C
Number of host bits	2	Always defined as number of binary 0s in mask
Number of subnet bits	6	32 – (network size + host size)
Number of subnets	$2^6 = 64$	$2^{\text{number-of-subnet-bits}}$
Number of hosts	$2^2 - 2 = 2$	$2^{\text{number-of-host-bits}} - 2$

Table D-67 contains the important binary calculations for finding the subnet number and subnet broadcast address, as summarized in Appendix E, RP-5A. To calculate the subnet number, perform a Boolean AND on the address and mask. To find the broadcast address for this subnet, change all the host bits to binary 1s in the subnet number. The host bits are in **bold** print in the table.

Table D-67 *Question 21: Binary Calculation of Subnet and Broadcast Addresses*

Address	192.168.15.230	`11000000 10101000 00001111 111001`**`10`**
Mask	255.255.255.252	`11111111 11111111 11111111 111111`**`00`**
AND result (subnet number)	192.168.15.228	`11000000 10101000 00001111 111001`**`00`**
Change host to 1s (broadcast address)	192.168.15.231	`11000000 10101000 00001111 111001`**`11`**

Just add 1 to the subnet number to get the first valid IP address; just subtract 1 from the broadcast address to get the last valid IP address. In this case:

192.168.15.229 through 192.168.15.230

Alternately, you can use the processes that only use decimal math to find the subnet and broadcast address. Table D-68 shows the work for this problem, with some explanation of the work following the table.

Table D-68 *Question 21: Subnet, Broadcast, First and Last Addresses Calculated Using Subnet Chart*

	Octet 1	Octet 2	Octet 3	Octet 4
Mask	255	255	255	252
Address	192	168	15	230
Subnet Number	192	168	15	228
First Valid Address	192	168	15	229
Last Valid Address	192	168	15	230
Broadcast	192	168	15	231

This subnetting scheme uses a difficult mask because one of the octets is not a 0 or a 255. The fourth octet is "interesting" in this case. The key part of the trick to get the right answers is to calculate the magic number, which is 256 – 252 = 4 in this case (256 – mask's value in the interesting octet). The subnet number's value in the interesting octet (inside the box) is the multiple of the magic number that is not higher than the original IP address's value in the interesting octet. In this case, 228 is the multiple of 4 that is closest to 230 but not higher than 230. So, the fourth octet of the subnet number is 228.

The second part of this process calculates the subnet broadcast address, with the tricky part, as usual, in the "interesting" octet. Take the subnet number's value in the interesting octet, add the magic number, and subtract 1. That is the broadcast address's value in the interesting octet. In this case, 228 + 4 – 1 = 231.

Answer to Problem 22 in Problem Set 5

Table D-69 *Question 22: Size of Network, Subnet, Host, Number of Subnets, Number of Hosts*

Item	Example	Rules to Remember
Address	10.1.1.1	—
Mask	255.248.0.0	—
Number of network bits	8	Always defined by Class A, B, C
Number of host bits	19	Always defined as number of binary 0s in mask
Number of subnet bits	5	32 – (network size + host size)
Number of subnets	$2^5 = 32$	$2^{\text{number-of-subnet-bits}}$
Number of hosts	$2^{19} - 2 = 524,286$	$2^{\text{number-of-host-bits}} - 2$

Table D-70 contains the important binary calculations for finding the subnet number and subnet broadcast address, as summarized in Appendix E, RP-5A. To calculate the subnet number, perform a Boolean AND on the address and mask. To find the broadcast address for this subnet, change all the host bits to binary 1s in the subnet number. The host bits are in **bold** print in the table.

Table D-70 *Question 22: Binary Calculation of Subnet and Broadcast Addresses*

Address	10.1.1.1	00001010 00000**001 00000001 00000001**
Mask	255.248.0.0	11111111 11111**000 00000000 00000000**
AND result (subnet number)	10.0.0.0	00001010 00000**000 00000000 00000000**
Change host to 1s (broadcast address)	10.7.255.255	00001010 00000**111 11111111 11111111**

Just add 1 to the subnet number to get the first valid IP address; just subtract 1 from the broadcast address to get the last valid IP address. In this case:

10.0.0.1 through 10.7.255.254

Take a closer look at the subnet part of the subnet address, as shown in bold here: 0000 1010 **0000 0**000 0000 0000 0000 0000. The subnet part of the address is all binary 0s, making this subnet a zero subnet.

Alternately, you can use the processes that only use decimal math to find the subnet and broadcast address. Table D-71 shows the work for this problem, with some explanation of the work following the table.

Table D-71 *Question 22: Subnet, Broadcast, First and Last Addresses Calculated Using Subnet Chart*

	Octet 1	Octet 2	Octet 3	Octet 4
Mask	255	248	0	0
Address	10	1	1	1
Subnet Number	10	0	0	0
First Valid Address	10	0	0	1
Last Valid Address	10	7	255	254
Broadcast	10	7	255	255

This subnetting scheme uses a difficult mask because one of the octets is not a 0 or a 255. The second octet is "interesting" in this case. The key part of the trick to get the right answers is to calculate the magic number, which is 256 – 248 = 8 in this case (256 – mask's value in the interesting octet). The subnet number's value in the interesting octet (inside the box) is the multiple of the magic number that is not higher than the original IP address's value in the interesting octet. In this case, 0 is the multiple of 8 that is closest to 1 but not higher than 1. So, the second octet of the subnet number is 0.

The second part of this process calculates the subnet broadcast address, with the tricky part, as usual, in the "interesting" octet. Take the subnet number's value in the interesting octet, add the magic number, and subtract 1. That is the broadcast address's value in the interesting octet. In this case, 0 + 8 – 1 = 7.

Answer to Problem 23 in Problem Set 5

Table D-72 *Question 23: Size of Network, Subnet, Host, Number of Subnets, Number of Hosts*

Item	Example	Rules to Remember
Address	172.16.1.200	—
Mask	255.255.240.0	—
Number of network bits	16	Always defined by Class A, B, C
Number of host bits	12	Always defined as number of binary 0s in mask
Number of subnet bits	4	32 – (network size + host size)
Number of subnets	$2^4 = 16$	$2^{\text{number-of-subnet-bits}}$
Number of hosts	$2^{12} - 2 = 4094$	$2^{\text{number-of-host-bits}} - 2$

Table D-73 contains the important binary calculations for finding the subnet number and subnet broadcast address, as summarized in Appendix E, RP-5A. To calculate the subnet number, perform a Boolean AND on the address and mask. To find the broadcast address for this subnet, change all the host bits to binary 1s in the subnet number. The host bits are in **bold** print in the table.

Table D-73 *Question 23: Binary Calculation of Subnet and Broadcast Addresses*

Address	172.16.1.200	10101100 00010000 0000**0001 11001000**
Mask	255.255.240.0	11111111 11111111 1111**0000 00000000**
AND result (subnet number)	172.16.0.0	10101100 00010000 0000**0000 00000000**
Change host to 1s (broadcast address)	172.16.15.255	10101100 00010000 0000**1111 11111111**

Just add 1 to the subnet number to get the first valid IP address; just subtract 1 from the broadcast address to get the last valid IP address. In this case:

172.16.0.1 through 172.16.15.254

Take a closer look at the subnet part of the subnet address, as shown in bold here: 1010 1100 0001 0000 **0000** 0000 0000 0000. The subnet part of the address is all binary 0s, making this subnet a zero subnet.

Alternately, you can use the processes that only use decimal math to find the subnet and broadcast address. Table D-74 shows the work for this problem, with some explanation of the work following the table.

Table D-74 *Question 23: Subnet, Broadcast, First and Last Addresses Calculated Using Subnet Chart*

	Octet 1	Octet 2	Octet 3	Octet 4
Mask	255	255	240	0
Address	172	16	1	200
Subnet Number	172	16	0	0
First Valid Address	172	16	0	1
Last Valid Address	172	16	15	254
Broadcast	172	16	15	255

This subnetting scheme uses a difficult mask because one of the octets is not a 0 or a 255. The third octet is "interesting" in this case. The key part of the trick to get the right answers is to calculate the magic number, which is 256 – 240 = 16 in this case (256 – mask's value in the interesting octet). The subnet number's value in the interesting octet (inside the box) is the multiple of the magic number that is not higher than the original IP address's value in the interesting octet. In this case, 0 is the multiple of 16 that is closest to 1 but not higher than 1. So, the third octet of the subnet number is 0.

The second part of this process calculates the subnet broadcast address, with the tricky part, as usual, in the "interesting" octet. Take the subnet number's value in the interesting octet, add the magic number, and subtract 1. That is the broadcast address's value in the interesting octet. In this case, 0 + 16 – 1 = 15.

Answer to Problem 24 in Problem Set 5

Table D-75 *Question 24: Size of Network, Subnet, Host, Number of Subnets, Number of Hosts*

Item	Example	Rules to Remember
Address	172.16.0.200	—
Mask	255.255.255.192	—
Number of network bits	16	Always defined by Class A, B, C
Number of host bits	6	Always defined as number of binary 0s in mask
Number of subnet bits	10	32 – (network size + host size)
Number of subnets	$2^{10} = 1024$	$2^{\text{number-of-subnet-bits}}$
Number of hosts	$2^6 - 2 = 62$	$2^{\text{number-of-host-bits}} - 2$

Table D-76 contains the important binary calculations for finding the subnet number and subnet broadcast address, as summarized in Appendix E, RP-5A. To calculate the subnet number, perform a Boolean AND on the address and mask. To find the broadcast address for this subnet, change all the host bits to binary 1s in the subnet number. The host bits are in **bold** print in the table.

Table D-76 *Question 24: Binary Calculation of Subnet and Broadcast Addresses*

Address	172.16.0.200	10101100 00010000 00000000 11**001000**
Mask	255.255.255.192	11111111 11111111 11111111 11**000000**
AND result (subnet number)	172.16.0.192	10101100 00010000 00000000 11**000000**
Change host to 1s (broadcast address)	172.16.0.255	10101100 00010000 00000000 11**111111**

Just add 1 to the subnet number to get the first valid IP address; just subtract 1 from the broadcast address to get the last valid IP address. In this case:

172.16.0.193 through 172.16.0.254

Alternately, you can use the processes that only use decimal math to find the subnet and broadcast address. Table D-77 shows the work for this problem, with some explanation of the work following the table.

This subnetting scheme uses a difficult mask because one of the octets is not a 0 or a 255. The fourth octet is “interesting” in this case. The key part of the trick to get the right

Table D-77 *Question 24: Subnet, Broadcast, First and Last Addresses Calculated Using Subnet Chart*

	Octet 1	Octet 2	Octet 3	Octet 4
Mask	255	255	255	192
Address	172	16	0	200
Subnet Number	172	16	0	192
First Valid Address	172	16	0	193
Last Valid Address	172	16	0	254
Broadcast	172	16	0	255

answers is to calculate the magic number, which is 256 – 192 = 64 in this case (256 – mask's value in the interesting octet). The subnet number's value in the interesting octet (inside the box) is the multiple of the magic number that is not higher than the original IP address's value in the interesting octet. In this case, 192 is the multiple of 64 that is closest to 200 but not higher than 200. So, the fourth octet of the subnet number is 192.

The second part of this process calculates the subnet broadcast address, with the tricky part, as usual, in the "interesting" octet. Take the subnet number's value in the interesting octet, add the magic number, and subtract 1. That is the broadcast address's value in the interesting octet. In this case, 192 + 64 – 1 = 255.

You can easily forget that the subnet part of this address, when using this mask, actually covers all of the third octet as well as 2 bits of the fourth octet. For instance, the valid subnet numbers in order are listed here:

172.16.0.0 (zero subnet)
172.16.0.64
172.16.0.128
172.16.0.192
172.16.1.0
172.16.1.64
172.16.1.128
172.16.1.192
172.16.2.0
172.16.2.64
172.16.2.128

172.16.2.192
172.16.3.0
172.16.3.64
172.16.3.128
172.16.3.192

And so on.

Answer to Problem 25 in Problem Set 5

Congratulations, you made it through all the extra subnetting practice! Here is an easy one to complete your review—one with no subnetting at all.

Table D-78 *Question 25: Size of Network, Subnet, Host, Number of Subnets, Number of Hosts*

Item	Example	Rules to Remember
Address	10.1.1.1	—
Mask	255.0.0.0	—
Number of network bits	8	Always defined by Class A, B, C
Number of host bits	24	Always defined as number of binary 0s in mask
Number of subnet bits	0	32 – (network size + host size)
Number of subnets	0	$2^{number-of-subnet-bits}$
Number of hosts	$2^{24} - 2 = 16,777,214$	$2^{number-of-host-bits} - 2$

Table D-79 contains the important binary calculations for finding the subnet number and subnet broadcast address, as summarized in Appendix E, RP-5A. To calculate the subnet number, perform a Boolean AND on the address and mask. To find the broadcast address for this subnet, change all the host bits to binary 1s in the subnet number. The host bits are in **bold** print in the table.

Table D-79 *Question 25: Binary Calculation of Subnet and Broadcast Addresses*

Address	10.1.1.1	00001010 **00000001 00000001 00000001**
Mask	255.0.0.0	11111111 **00000000 00000000 00000000**
AND result (subnet number)	10.0.0.0	00001010 **00000000 00000000 00000000**
Change host to 1s (broadcast address)	10.255.255.255	00001010 **11111111 11111111 11111111**

Just add 1 to the subnet number to get the first valid IP address; just subtract 1 from the broadcast address to get the last valid IP address. In this case:

10.0.0.1 through 10.255.255.254

Alternately, you can use the processes that only use decimal math to find the subnet and broadcast address. Table D-80 shows the work for this problem, with some explanation of the work following the table.

Table D-80 *Question 25: Subnet, Broadcast, First and Last Addresses Calculated Using Subnet Chart*

	Octet 1	Octet 2	Octet 3	Octet 4
Mask	255	0	0	0
Address	10	1	1	1
Network Number	10	0	0	0
First Valid Address	10	0	0	1
Last Valid Address	10	255	255	254
Broadcast	10	255	255	255

Answers to Problem Set 6

This section includes the answers to the three problems listed in Problem Set 6.

Answer to Problem 1 in Problem Set 6

The answer is as follows:

- 172.32.0.0 (zero subnet)
- 172.32.4.0
- 172.32.8.0
- 172.32.12.0
- 172.32.16.0
- 172.32.20.0
- 172.32.24.0

(skipping many subnets; each new subnet is the same as the previous subnet, after adding 4 to the third octet)

- 172.32.248.0
- 172.32.252.0 (broadcast subnet)

The process to find all subnets depends on three key pieces of information, as follows:

- The mask has fewer than 8 subnet bits (6 bits), because the network is a Class B network (16 network bits), and the mask has 22 binary 1s in it—implying 10 host bits, and leaving 6 subnet bits.
- The mask in dotted decimal format is 255.255.252.0. The interesting octet is the third octet because the mask's value in the third octet (252) is the only mask octet that is not a 255 or a 0.
- Each successive subnet number is 4 higher than the previous subnet number, in the interesting octet, because the magic number is 256 – 252 = 4.

As a result, in this case, all the subnets begin with 172.32, have a multiple of 4 in the third octet, and end in 0.

Table D-81 shows the results of the various steps of the process, as outlined in Appendix E, RP-7A.

Table D-81 *Problem Set 6, Question 1: Answer Table*

	Octet 1	Octet 2	Octet 3	Octet 4
Subnet Mask (Step 1)	255	255	252	0
Magic Number (Step 3)			256 – 252 = 4	
Zero Subnet Number (Step 4)	172	32	0	0
Next Subnet (Step 5)	172	32	4	0
Next Subnet (Step 5)	172	32	8	0
Next Subnet (Step 5)	172	32	12	0
Next Subnet (Step 5)	172	32	16	0
(You may need many more such rows…)	172	32	X	0
Next Subnet	172	32	244	0
Next Subnet (Step 5)	172	32	248	0
Broadcast Subnet (Step 6)	172	32	252	0
Out of Range—Stop Process (Step 6)	172	32	256	0

Answer to Problem 2 in Problem Set 6

The answer is as follows:

- 200.1.2.0 (zero subnet)
- 200.1.2.16
- 200.1.2.32
- 200.1.2.48
- 200.1.2.64
- 200.1.2.80

  (skipping many subnets; each new subnet is the same as the previous subnet, after adding 16 to the fourth octet)
- 200.1.2.224
- 200.1.2.240 (broadcast subnet)

The process to find all subnets depends on three key pieces of information, as follows:

- The mask has fewer than 8 subnet bits (4 bits), because the network is a Class C network (24 network bits), and the mask has 28 binary 1s in it—implying 4 host bits, and leaving 4 subnet bits.
- The mask in dotted decimal format is 255.255.255.240. The interesting octet is the fourth octet because the mask's value in the fourth octet (240) is the only mask octet that is not a 255 or a 0.
- Each successive subnet number is 16 higher than the previous subnet number, in the interesting octet, because the magic number is 256 – 240 = 16.

As a result, in this case, all the subnets begin with 200.1.2 and have a multiple of 16 in the fourth octet.

Table D-82 shows the results of the various steps of the process, as outlined in Appendix E, RP-7A.

Table D-82 *Problem Set 6, Question 2: Answer Table*

	Octet 1	Octet 2	Octet 3	Octet 4
Subnet Mask (Step 1)	255	255	255	240
Magic Number (Step 3)				256 – 240 = 16
Zero Subnet Number (Step 4)	200	1	2	0
Next Subnet (Step 5)	200	1	2	16
Next Subnet (Step 5)	200	1	2	32
Next Subnet (Step 5)	200	1	2	48
(You may need many more such rows…) (Step 5)	200	1	2	X
Next Subnet (Step 5)	200	1	2	224
Broadcast Subnet (Step 6)	200	1	2	240
Out of Range—Stop Process (Step 6)	200	1	2	256

Answer to Problem 3 in Problem Set 6

The answer is as follows:

- 10.0.0.0 (zero subnet)
- 10.2.0.0
- 10.4.0.0
- 10.6.0.0

  (skipping many subnets; each new subnet is the same as the previous subnet, after adding 2 to the second octet)
- 10.252.0.0
- 10.254.0.0 (broadcast subnet)

The process to find all subnets depends on three key pieces of information, as follows:

- The mask has fewer than 8 subnet bits (7 subnet bits), because the network is a Class A network (8 network bits), and the mask has 15 binary 1s in it—implying 17 host bits, and leaving 7 subnet bits.

- The mask in dotted decimal format is 255.254.0.0. The interesting octet is the second octet because the mask's value in the second octet (254) is the only mask octet that is not a 255 or a 0.
- Each successive subnet number is 2 higher than the previous subnet number, in the interesting octet, because the magic number is 256 – 254 = 2.

As a result, in this case, all the subnets begin with 10, have a multiple of 2 in the second octet, and end in 0.0.

Table D-83 shows the results of the various steps of the process, as outlined in Appendix E, RP-7A.

Table D-83 *Problem Set 6, Question 3: Answer Table*

	Octet 1	Octet 2	Octet 3	Octet 4
Subnet Mask (Step 1)	255	254	0	0
Magic Number (Step 3)		256 – 254 = 2		
Zero Subnet Number (Step 4)	10	0	0	0
Next Subnet (Step 5)	10	2	0	0
Next Subnet (Step 5)	10	4	0	0
Next Subnet (Step 5)	10	6	0	0
(You may need many more such rows…) (Step 5)	10	X	0	0
Next Subnet (Step 5)	10	252	0	0
Broadcast Subnet (Step 6)	10	254	0	0
Out of Range—Stop Process (Step 6)	10	256	0	0

Answers to Problem Set 7

This section includes the answers to the three problems listed in Problem Set 7. The first two problems happen to use more than 8 subnet bits, so the explanations follow the process summarized in Appendix E, RP-7B. Problem 3 uses exactly 8 subnet bits, which makes for a simpler answer, as described in Chapter 12 of the *CCENT/CCNA ICND1 Official Exam Certification Guide* (Appendix H in the *CCNA ICND2 Official Exam Certification Guide*). Appendix E does not have a reference page specifically for the case with exactly 8 subnet bits.

Answer to Problem 1 in Problem Set 7

This problem has a 9-bit subnet field, meaning that 2^9 or 512 possible subnets exist. The following list shows some of the subnets, which should be enough to see the trends in how to find all subnet numbers:

- 172.32.0.0 (zero subnet)
- 172.32.0.128
- 172.32.1.0
- 172.32.1.128
- 172.32.2.0
- 172.32.2.128
- 172.32.3.0
- 172.32.4.128

  (skipping many subnets; each new subnet is the same as the previous subnet, after adding 4 to the third octet)

- 172.32.254.0
- 172.32.254.128
- 172.32.255.0
- 172.32.255.128 (broadcast subnet)

The process to find all subnets depends on three key pieces of information, as follows:

- The mask has more than 8 subnet bits (9 bits), because the network is a Class B network (16 network bits), and the mask has 25 binary 1s in it—implying 7 host bits, and leaving 9 subnet bits.
- The mask in dotted decimal format is 255.255.255.128. The interesting octet is the fourth octet because the mask's value in the fourth octet (128) is the only mask octet that is not a 255 or a 0.
- The magic number, which will be used to calculate each successive subnet number, is 256 – 128 = 128.

This problem uses the process summarized in Appendix E, RP-7B, which actually extends process RP-7A. Essentially, you follow the same details in process RP-7A until that process creates a sum of 256, at which point you then follow the instructions in process RP-7B.

To make sure this process is clear, Table D-84 shows the work in progress using process RP-7A, up to the point that a sum of 256 occurs. (Note that the last row is not the correct value—it is just there to make a point about how the process works.)

Table D-84 *Problem Set 7, Problem 1: Process up to the First Use of RP-7B*

	Octet 1	Octet 2	Octet 3	Octet 4
Subnet Mask (Step 1)	255	255	255	128
Magic Number (Step 3)				256 – 128 = 128
Zero Subnet Number (Step 4)	172	32	0	0
Next Subnet (Step 5)	172	32	0	128
RP-7B Step 6 Needs to Be Used Here (sum of 256 in the 4^{th} octet)	172	32	0	256*

The last row in the table shows the sum of 256 in the interesting octet, which means that RP-7B Step 6 should instead be used. In this case, instead of writing a 256 in the fourth octet, the following should occur:

Step 6a: Write down a 0 in the interesting (fourth) octet.

Step 6b: Add 1 to the previous subnet's value in the octet to the left (third octet).

Table D-85 shows the new and correct result for this next subnet.

Table D-85 *Problem Set 7, Problem 1: Correct Use of RP-7B Step 6*

	Octet 1	Octet 2	Octet 3	Octet 4
Subnet Mask (Step 1)	255	255	255	128
Magic Number (Step 3)				256 – 128 = 128
Zero Subnet Number (Step 4)	172	32	0	0
Next Subnet (Step 5)	172	32	0	128
Work Done at RP-7B Step 6 for Next Subnet			above value (0) plus 1	write a 0 below
RP-7B Step 6 Needs to Be Used Here (sum of 256 in the 4^{th} octet)	172	32	1	0

Continuing this process, each time the fourth octet adds up to 256, instead write down a 0 in that octet, and then add 1 to the previous subnet's third octet. Essentially, it is like carrying a 1 in basic decimal addition.

The key to how to stop the process, and find the broadcast subnet, is listed as part of RP-7B Step 7. In short, when this process would cause you to carry a 1 to the octet to the left, which would then in turn change the value in one of the octets in the network portion of the number, you should stop. Table D-86 shows the work in progress that leads up to this point.

Table D-86 *Problem Set 7, Problem 1: Finding Where to Stop Using RP-7B Step 7*

	Octet 1	Octet 2	Octet 3	Octet 4
Subnet Mask (Step 1)	255	255	255	128
Magic Number (Step 3)				256 – 128 = 128
(Many subnets not listed here…)				
Next Subnet (Step 5)	172	32	255	0
Next Subnet (Step 5)	172	32	255	128
Work Done at RP-7B Step 6 for Next Subnet		previous octet (32) plus 1	It would be 256—write a 0, and carry a 1	It would be 256—write a 0, carry a 1
Next Number If You Continue Adding	172	33	0	0

According to the process, any time you would need to write down a 256, do not—instead, write a 0, and carry a 1 to the octet to the left. You start by adding the magic number (128) in the interesting octet (fourth octet), which adds up to 256. This triggers the process by which you instead write a 0 and carry a 1 to the octet to the left (third octet). However, in this case, the sum in the third octet also adds up to 256, meaning that you should write a 0, and carry a 1 to the second octet. However, because a Class B network is in use, this would change the value in one of the network octets. So, you know to stop work at this point. The previous subnet, 172.32.255.128 in this case, is the broadcast subnet.

Answer to Problem 2 in Problem Set 7

This problem has a 13-bit subnet field, meaning that 2^{13} or 8192 possible subnets exist. The following list shows some of the subnets, which should be enough to see the trends in how to find all subnet numbers:

- 10.0.0.0 (zero subnet)
- 10.0.8.0
- 10.0.16.0
- 10.0.24.0

  (skipping several subnets…)

- 10.0.248.0
- 10.1.0.0
- 10.1.8.0
- 10.1.16.0

  (skipping several subnets…)
- 10.1.248.0
- 10.2.0.0
- 10.2.8.0
- 10.2.16.0

  (skipping several subnets…)
- 10.255.232.0
- 10.255.240.0
- 10.255.248.0 (broadcast subnet)

The process to find all subnets depends on three key pieces of information, as follows:

- The mask has more than 8 subnet bits (13 bits), because the network is a Class A network (8 network bits), and the mask has 21 binary 1s in it—implying 11 host bits, and leaving 13 subnet bits.
- The mask in dotted decimal format is 255.255.248.0. The interesting octet is the third octet because the mask's value in the third octet (248) is the only mask octet that is not a 255 or a 0.
- The magic number, which will be used to calculate each successive subnet number, is 256 – 248 = 8.

This problem uses the process summarized in Appendix E, RP-7B, which actually extends process RP-7A. Essentially, you follow the same details in process RP-7A until that process creates a sum of 256, at which point you then follow the instructions in process RP-7B. To make sure this process is clear, Table D-87 shows the work in progress using process RP-7A, up to the point that a sum of 256 occurs. (Note that the last row is not the correct value—it is just there to make a point about how the process works.)

Table D-87 *Problem Set 7, Problem 2: Process up to the First Use of RP-7B*

	Octet 1	Octet 2	Octet 3	Octet 4
Subnet Mask (Step 1)	255	255	248	0
Magic Number (Step 3)			256 – 248 = 8	
Zero Subnet Number (Step 4)	10	0	0	0
Next Subnet (Step 5)	10	0	8	0
(Skipping several subnets…)				
Next Subnet (Step 5)	10	0	248	0
RP-7B Step 6 Needs to Be Used Here (sum of 256 in the 3rd octet)	10	0	256	0

The last row in the table shows the sum of 256 in the interesting octet, which means that RP-7B Step 6 should instead be used. In this case, instead of writing a 256 in the third octet, the following should occur:

Step 6A: Write down a 0 in the interesting (third) octet.

Step 6B: Add 1 to the previous subnet's value in the octet to the left (second octet).

Table D-88 shows the new and correct result for this next subnet.

Table D-88 *Problem Set 7, Problem 2: Correct Use of RP-7B Step 6*

	Octet 1	Octet 2	Octet 3	Octet 4
Subnet Mask (Step 1)	255	255	248	0
Magic Number (Step 3)			256 – 248 = 8	
Zero Subnet Number (Step 4)	10	0	0	0
Next Subnet (Step 5)	10	0	8	0
(Skipping several subnets…)				
Next Subnet (Step 5)	10	0	248	0
Work Done at RP-7B Step 6 for Next Subnet		above value (0) plus 1	write a 0 below	
RP-7B Step 6 Needs to Be Used Here (sum of 256 in the 3rd octet)	10	1	0	0

Continuing this process, each time the third octet adds up to 256, instead write down a 0 in that octet, and then add 1 to the previous subnet's second octet. Essentially, it is like carrying a 1 in basic decimal addition.

The key to how to stop the process, and find the broadcast subnet, is listed as part of RP-7B Step 7. In short, when this process would cause you to carry a 1 to the octet to the left, which would then in turn change the value in one of the octets in the network portion of the number, you should stop. Table D-89 shows the work in progress that leads up to this point.

Table D-89 *Problem Set 7, Problem 2: Finding Where to Stop Using RP-7B Step 7*

	Octet 1	Octet 2	Octet 3	Octet 4
Subnet Mask (Step 1)	255	255	255	128
Magic Number (Step 3)			256 – 248 = 8	
(Many subnets not listed here…)				
Next Subnet (Step 5)	10	255	240	0
Next Subnet (Step 5)	10	255	248	0
Work Done at RP-7B Step 6 for Next Subnet	Adding 1 here would change the network number	255 + 1 = 256—write a 0, and carry a 1	248 + 8 = 256—write a 0, carry a 1	
Next Number If You Continue Adding	11	0	0	0

According to the process, any time you would need to write down a 256, do not—instead, write a 0, and carry a 1 to the octet to the left. You start by adding the magic number (8) in the interesting octet (third octet), which adds up to 256. This triggers the process by which you instead write a 0 and carry a 1 to the octet to the left (second octet). However, in this case, the sum in the second octet also adds up to 256, meaning that you should write a 0, and carry a 1 to the first octet. However, because a Class A network is in use, this would change the value in the one network octet. So, you know to stop work at this point. The previous subnet, 10.255.248.0 in this case, is the broadcast subnet.

Answer to Problem 3 in Problem Set 7

This problem has an 8-bit subnet field, meaning that 2^8 or 256 possible subnets exist. The following list shows some of the subnets, which should be enough to see the trends in how to find all subnet numbers:

- 172.20.0.0 (zero subnet)
- 172.20.1.0

- 172.20.2.0
- 172.20.3.0
- 172.20.4.0

  (skipping many subnets; each new subnet is the same as the previous subnet, after adding 1 to the third octet)

- 172.20.252.0
- 172.20.253.0
- 172.20.254.0
- 172.20.255.0 (broadcast subnet)

The process to find all subnets depends on three key pieces of information, as follows:

- The mask has exactly 8 subnet bits, specifically all bits in the third octet, making the third octet the interesting octet.
- The magic number is 256 – 255, because the mask's value in the interesting (third) octet is 255.
- Beginning with the network number of 172.20.0.0, which is the same value as the zero subnet, just add the magic number (1) in the interesting octet.

Essentially, you just count by 1 in the third octet, until you reach the highest legal number (255). The first subnet, 172.20.0.0, is the zero subnet, and the last subnet, 172.20.255.0, is the broadcast subnet.

Answers to Problem Set 8

This section includes the answers to the two problems listed in Problem Set 8. The explanations included here follow the summarized process listed as RP-8 in Appendix E.

Answer to Problem 1 in Problem Set 8

The answer is 172.30.16.0/23.

The first step in the process for finding the answer is to analyze the existing subnets. It is also particularly useful to list the values in numeric order. Table D-90 lists the existing subnets, as stated in the problem statement, along with the range of addresses and broadcast address in each subnet.

Table D-90 *Existing Subnets, Range of Addresses, and Broadcast Addresses*

	Subnet Number	First Address	Last Address	Broadcast Address
1st	172.30.0.0/20	172.30.0.1	172.30.15.254	172.30.15.255
2nd	172.30.20.0/22	172.30.20.1	172.30.23.254	172.30.23.255
3rd	172.30.32.0/25	172.30.32.1	172.30.32.126	172.30.32.127
4th	172.30.34.0/30	172.30.34.1	172.30.34.2	172.30.34.3
5th	172.30.34.4/30	172.30.34.5	172.30.34.6	172.30.34.7
6th	172.30.34.8/30	172.30.34.9	172.30.34.10	172.30.34.11
7th	172.30.34.12/30	172.30.34.13	172.30.34.14	172.30.34.15
8th	172.30.34.16/30	172.30.34.17	172.30.34.18	172.30.34.19

The following steps explain the work done at each step of Appendix E, RP-8, to find the answer to this problem:

Step 1 Calculate the ranges of addresses in each subnet, as listed in Table D-90.

Step 2 The problem stated that the new subnet must accommodate 300 hosts. To do so, the mask must have at least 9 host bits, because $2^8 - 2 = 254$, but $2^9 - 2 = 510$. The mask value that ends with nine binary 0s is 11111111 11111111 11111110 00000000, or 255.255.254.0, or /23.

Step 3 Assuming a mask of /23, in network 172.30.0.0, the possible subnets would be as follows:

- 172.30.0.0 (zero subnet)
- 172.30.2.0
- 172.30.4.0
- 172.30.6.0
- 172.30.8.0
- 172.30.10.0
- 172.30.12.0
- 172.30.14.0
- 172.30.16.0
- 172.30.18.0

And so on, with each successive subnet found by adding 2 to the third octet.

Step 4 For step 4, you should calculate the range of IP addresses in each possible subnet found at Step 3. Although Table D-91 lists those values, you may have noticed that several of the subnet numbers listed at Step 3 obviously overlap with existing subnet 172.30.0.0/20 (see the range of addresses in Table D-90.) So, you could save some effort and ignore those subnets. Note that Table D-91 does not list all the subnets, but it does include enough information to find the correct answer to this problem.

Table D-91 *Possible Subnets of 172.30.0.0, Mask 255.255.254.0*

	Subnet Number	First Address	Last Address	Broadcast Address
1st	172.30.0.0	172.30.0.1	172.30.1.254	172.30.1.255
2nd	172.30.2.0	172.30.2.1	172.30.3.254	172.30.3.255
3rd	172.30.4.0	172.30.4.1	172.30.5.254	172.30.5.255
4th	172.30.6.0	172.30.6.1	172.30.7.254	172.30.7.255
5th	172.30.8.0	172.30.8.1	172.30.9.254	172.30.9.255
6th	172.30.10.0	172.30.10.1	172.30.11.254	172.30.11.255
7th	172.30.12.0	172.30.12.1	172.30.13.254	172.30.13.255
8th	172.30.14.0	172.30.14.1	172.30.15.254	172.30.15.255
9th	172.30.16.0	172.30.16.1	172.30.17.254	172.30.17.255
10th	172.30.18.0	172.30.18.1	172.30.19.254	172.30.19.255

Step 5 Step 5 requires that you compare the two tables (D-90 and D-91 in this case). The first existing subnet—subnet 172.30.0.0/20—overlaps with the first eight subnets listed in Table D-91, so you draw a line through the first eight subnets in Table D-91. However, the ninth subnet in Table D-91—172.30.16.0/23—does not overlap with any of the existing subnets listed in Table D-90.

Step 6 You could continue analyzing the contents of the tables as suggested for Step 5, but because the problem statement asked for the smallest new subnet number, you already know the correct answer—172.30.16.0/23.

Answer to Problem 2 in Problem Set 8

The answer is 192.168.1.112/28.

The first step in the process for finding the answer is to analyze the existing subnets. It is also particularly useful to list the values in numeric order. Table D-92 lists the existing subnets, as stated in the problem statement, along with the range of addresses and broadcast address in each subnet.

Table D-92 *Existing Subnets, Range of Addresses, and Broadcast Addresses*

	Subnet Number	First Address	Last Address	Broadcast Address
1st	192.168.1.64/30	192.168.1.65	192.168.1.66	192.168.1.67
2nd	192.168.1.72/30	192.168.1.73	192.168.1.74	192.168.1.75
3rd	192.168.1.76/30	192.168.1.77	192.168.1.78	192.168.1.79
4th	192.168.1.128/26	192.168.1.129	192.168.1.190	192.168.1.191
5th	192.168.1.192/26	192.168.1.193	192.168.1.254	192.168.1.255

The following steps explain the work done at each step of Appendix E, RP-8, to find the answer to this problem.

Step 1 Calculate the ranges of addresses in each subnet, as listed in Table D-92.

Step 2 The problem stated that the new subnet must accommodate 13 hosts. To do so, the mask must have at least 4 host bits, because $2^3 - 2 = 6$, but $2^4 - 2 = 14$. The mask value that ends with four binary 0s is 11111111 11111111 11111111 11110000, or 255.255.255.240, or /28.

Step 3 Assuming a mask of /28, in network 192.168.1.0, the possible subnets would be as follows:

- 192.168.1.0 (zero subnet)
- 192.168.1.16
- 192.168.1.32
- 192.168.1.48
- 192.168.1.64
- 192.168.1.80
- 192.168.1.96
- 192.168.1.112
- 192.168.1.128
- 192.168.1.144

- 192.168.1.160
- 192.168.1.176
- 192.168.1.192
- 192.168.1.208
- 192.168.1.224
- 192.168.1.240 (broadcast subnet)

Step 4 For Step 4, you should calculate the range of IP addresses in each possible subnet found at Step 3. Table D-93 lists those values.

Table D-93 *Possible Subnets of 192.168.1.0, Mask 255.255.255.240*

	Subnet Number	First Address	Last Address	Broadcast Address
1st	192.168.1.0	192.168.1.1	192.168.1.14	192.168.1.15
2nd	192.168.1.16	192.168.1.17	192.168.1.30	192.168.1.31
3rd	192.168.1.32	192.168.1.33	192.168.1.46	192.168.1.47
4th	192.168.1.48	192.168.1.49	192.168.1.62	192.168.1.63
5th	192.168.1.64	192.168.1.65	192.168.1.78	192.168.1.79
6th	192.168.1.80	192.168.1.81	192.168.1.94	192.168.1.95
7th	192.168.1.96	192.168.1.97	192.168.1.110	192.168.1.111
8th	192.168.1.112	192.168.1.113	192.168.1.126	192.168.1.127
9th	192.168.1.128	192.168.1.129	192.168.1.142	192.168.1.143
10th	192.168.1.144	192.168.1.145	192.168.1.158	192.168.1.159
11th	192.168.1.160	192.168.1.161	192.168.1.174	192.168.1.175
12th	192.168.1.176	192.168.1.177	192.168.1.190	192.168.1.191
13th	192.168.1.192	192.168.1.193	192.168.1.206	192.168.1.207
14th	192.168.1.208	192.168.1.209	192.168.1.222	192.168.1.223
15th	192.168.1.224	192.168.1.225	192.168.1.238	192.168.1.239
16th	192.168.1.240	192.168.1.241	192.168.1.254	192.168.1.255

Step 5 Step 5 requires that you compare the two tables (D-92 and D-93 in this case), and draw a line through any of the possible new subnets (Table D-93) that overlap with the existing subnets. Table D-94 shows the revised table, with the overlapping subnets noted.

Table D-94 *Subnets of 192.168.1.0, Mask 255.255.255.240, that Do Not Overlap*

	Subnet Number	First Address	Last Address	Broadcast Address
1st	192.168.1.0	192.168.1.1	192.168.1.14	192.168.1.15
2nd	192.168.1.16	192.168.1.17	192.168.1.30	192.168.1.31
3rd	192.168.1.32	192.168.1.33	192.168.1.46	192.168.1.47
4th	192.168.1.48	192.168.1.49	192.168.1.62	192.168.1.63
5th	~~192.168.1.64~~	~~192.168.1.65~~	~~192.168.1.78~~	~~192.168.1.79~~
6th	192.168.1.80	192.168.1.81	192.168.1.94	192.168.1.95
7th	192.168.1.96	192.168.1.97	192.168.1.110	192.168.1.111
8th	192.168.1.112	192.168.1.113	192.168.1.126	192.168.1.127
9th	~~192.168.1.128~~	~~192.168.1.129~~	~~192.168.1.142~~	~~192.168.1.143~~
10th	~~192.168.1.144~~	~~192.168.1.145~~	~~192.168.1.158~~	~~192.168.1.159~~
11th	~~192.168.1.160~~	~~192.168.1.161~~	~~192.168.1.174~~	~~192.168.1.175~~
12th	~~192.168.1.176~~	~~192.168.1.177~~	~~192.168.1.190~~	~~192.168.1.191~~
13th	~~192.168.1.192~~	~~192.168.1.193~~	~~192.168.1.206~~	~~192.168.1.207~~
14th	~~192.168.1.208~~	~~192.168.1.209~~	~~192.168.1.222~~	~~192.168.1.223~~
15th	~~192.168.1.224~~	~~192.168.1.225~~	~~192.168.1.238~~	~~192.168.1.239~~
16th	~~192.168.1.240~~	~~192.168.1.241~~	~~192.168.1.254~~	~~192.168.1.255~~

Step 6 Notice that the eighth subnet listed in the table is the largest subnet that does not overlap with the existing subnet. So, the correct answer is 192.168.1.112/28.