TCP/IP subnetting cheatsheet

Send corrections to support@boson.com

Special Addressing Structures that are invalid and not used:

Special Stuff	Network	Host		Invalid Addresses
This Host (internal)	All Zeros	All Zeros	=	0.0.0.0
Network ID	This Net	All Zeros	=	X.X.X.0
Local Broadcast	All Ones	All Ones	=	255.255.255.255
Directed Broadcast	This Net	All Ones	=	X.X.X.255
Loopback (diag)	127	Anything	=	127.X.X.X

*** RFC 1166 [Reserved] Internet Numbers July 1990**

Internet Address	Network	Reference
10.X.X.X	Reserved	
172.(16-31).X.X	Reserved	
192.0.0.X	Reserved	JBP
192.0.1.X	Backbone-Test-C	RH6
192.0.2.X	Internet-Test-C	JBP
192.0.(3-255).X	Unassigned	NIC
192.1.(0-1).X	Backbone Local Nets	SGC
192.1.2.X	Backbone Fiber Nets	SGC
192.1.3.X	Backbone Apollo Nets	SGC

RFC 952 states you cannot use the underscore ("_") character in a DNS name.

An Octet is a binary number of 8 bits. The smallest number is '00000000' and the largest is '11111111'

A shortcut is: $2^{8\,(-1).}$ This power of 2 is the largest octet (all 8 bits are 1's). It is 255 in decimal.

High Order Bits (First Octet) = Network Class type:

Internet	Start		End		Microsoft	Range	Microsoft bits
Class A =	**00000000**	**-**	**01111111**	**=**	Class A =	**1-126**	**0**
Class B =	**10000000**	**-**	**00111111**	**=**	Class B =	**128-191**	**10**
Class C =	**11000000**	**-**	**00011111**	**=**	Class C =	**192-223**	**110**
Class D =	**11100000**	**-**	**00001111**	**=**	Class D =	**invalid**	**invalid**
Class E =	**11110000**	**-**	**00000111**	**=**	Class E =	**invalid**	**invalid**

In the above examples the high order bits for Class A (0), therefore leaving 7 bits remain for the Network ID **01111111**, which is 2 to the 7th power minus 2, remembering that all ones and all zeros are invalid.
An octet is based on 2 to-the-power-of 8, minus one or more bits.

To compute a Class A using power of 8: $2^{(8-2)-(128-2)}$ **1 - 126** Unique bits: '0'

For the Class B, make the 2nd bit in the network range 0, which is 1011, then add all bits **10111111**, which equals 191: $2^{(8-2)-(64-2)-1}$ **128 - 191** Unique bits: '10'

For the Class C, make the 3rd bit in the network range 0, which is 1101, then add all bits **11011111**, which equals 223: $2^{(8-2)-(32-2)-1}$ **192 - 223** Unique bits: '110'

Network ID's: Internet			**Network**
Class A	**01111111.**00000000.00000000.00000000	= 2 to the 7th power -2 = 126	net.h.h.h
Class B	**10111111.11111111.**00000000.00000000	= 2 to the 14th power = 16384	net.net.h.h
Class C	**11011111.11111111.11111111.**00000000	= 2 to the 21st power = 2097152	net.net.net.h

Host ID's: Internet			**Host**
Class A	00000000.**11111111.11111111.11111111**	= 2 to the 24th power - 2 = 16777214	x.255.255.255
Class B	00000000.00000000.**11111111.11111111**	= 2 to the 16th power -2 = 65534	x.y.255.255
Class C	00000000.00000000.00000000.**11111111**	= 2 to the 8th power - 2 = 254	x.y.z.255

Address Class Summary

Internet		Number of Networks		Number of Host Per Network		Range of Network ID's (First Octet)	
	Class A		126		16,777,214		1 -- 126	
	Class B		16,384		65,534		128 -- 191	
	Class C		2,097,152		254		192 -- 223	

Subnet masks usually consist of adjacent, high-order bits. This means there is always an equal number of host bits per subnet (network) bits… And there are only 7 (or 5) valid combinations! (More on this later…)

Masks			SubNets			Valid?
Binary Network	Decimal Mask		Binary Hosts	Bits Used	Number of Subnets	Valid Host Ranges
00000000	0	<--->	00000000	0	Net ID	Not a Sub
10000000	128*	<--->	00000001	1	Invalid. Need 2 bits.	
11000000	192	<--->	00000011	2	2	OK 1/7
11100000	224	<--->	00000111	3	6	OK 2/7
11110000	240	<--->	00001111	4	14	OK 3/7
11111000	248	<--->	00011111	5	30	OK 4/7
11111100	252	<--->	00111111	6	62	OK 5/7
11111110	254*	<--->	01111111	7	126	OK 6/7 *
11111111	255*	<--->	11111111	8	254	OK 7/7 *

* = 7 subs valid for Class A or B. Class C has only 5 valid - the last 2 are not used.

Binary-to-Decimal Conversion Table: MEMORIZE

128	64	32	16	8	4	2	1
0/1	0/1	0/1	0/1	0/1	0/1	0/1	0/1

Sample Test Question: You are given a Network ID of 132.7.x.x and need 5 SubNets…

1. Memorize 2bits=2subs, 3bits=6subs, 4bits=14subs, 5bits=30subs, 6bits=62subs.
2. List the additional octet added to the default subnet mask in decimal notation (flip host bits with network).
   Note: Taking subnets needed in binary, and flipping with net bits to determine mask, does not always work.
3. Convert the rightmost 1 bit of this value to decimal notation, which is the incremental value between each subnet value, known as "Delta". Take this bit (00100000) and add to the left (11100000); you get 3 bits.
4. The number of subnet network ID's that are created is Two less than to the power of n, where n is the number of bits used it Step 2. In other words, the # of subnets = (2^n)-2) = ((2^3)-2) = ((8)-2) = 6.
5. Append "Delta" to the original network ID to give the first subnet network ID (00100000) = 32.
6. Repeat Step 4 for each subnet network ID, incrementing each successive value by "Delta".
7. Since you can always pretend it is a Class C, you know you need 3 bits (11100000) for 6 subnets.
8. Since the rightmost bit (00100000) is 32, this is the incremental value (aka Delta), and is your starting point.
9. There are (2^3)-2=6 subnets created. The -2 is important, because you cannot use all 0 or all 1 subnets.

So, the answers to Step 8 are:

Bit 1/8: 32x0 = All 0, invalid.		Invalid because you cannot multiply by 0 (all 0) and have it ever be valid.
Bit 2/8: 32x1 = 32	Delta 1/6	First number 32 and last number 63 invalid. This leaves 33-62 valid.
Bit 3/8: 32x2 = 64	Delta 2/6	First number 64 and last number 95 invalid. This leaves 65-94 valid.
Bit 4/8: 32x3 = 96	Delta 3/6	First number 96 and last number 127 invalid. This leaves 97-126 valid.
Bit 5/8: 32x4 = 128	Delta 4/6	First number 128 and last number 159 invalid. This leaves 129-158 valid.
Bit 6/8: 32x5 = 160	Delta 5/6	First number 160 and last number 191 invalid. This leaves 161-190 valid.
Bit 7/8: 32x6 = 192	Delta 6/6	First number 192 and last number 223 invalid. This leaves 193-222 valid.
Bit 8/8: 32x7 = All 1, invalid.		Invalid because you cannot multiply by 8 (all 1) and have it ever be valid.

Notes:

The special all 0 address is the network ID for that subnet.
The special all 1 address is the network broadcast for that subnet
You must always use at least 2 bits for any subnet and/or network ID to be valid.
The first numbers in table ranges are always all 1's and last numbers are always all 0's.
You can Not use the Delta bit x 0 or Delta bit x 8 range. This gives you a network ID of all 0 or all 1.

Network ID Classes A, B, C. (Long-Form)

*** All SubNets can be broken down to: 7 ranges, 3 classes each (A, B, C)**

	Max. No. Subnets	Math For Host ID	Max. Host per Subnet	Subnet Mask	Sub Bits Required	Host Bits Required	
Class A	0	2^{24-2}	16,777,214	255.0.0.0	0	24	**Class A**
	Invalid	2^{23-2}	Invalid	128 = invalid	1	23	
	2	2^{22-2}	4,194,302	255.192.0.0	2	22	Range 1/7
	6	2^{21-2}	2,097,150	255.224.0.0	3	21	Range 2/7
	14	2^{20-2}	1,048,574	255.240.0.0	4	20	Range 3/7
	30	2^{19-2}	524,286	255.248.0.0	5	19	Range 4/7
	62	2^{18-2}	262,142	255.252.0.0	6	18	Range 5/7
	126	2^{17-2}	131,070	255.254.0.0	7	17	Range 6/7
Class B	0	2^{16-2}	65,534	255.255.0.0	8	16	A' Range 7/7 & Start B
	Invalid	2^{15-2}	Invalid	128 = invalid	9	15	
	2	2^{14-2}	16,382	255.255.192.0	10	14	Range 1/7
	6	2^{13-2}	8,190	255.255.224.0	11	13	Range 2/7
	14	2^{12-2}	4,094	255.255.240.0	12	12	Range 3/7
	30	2^{11-2}	2,046	255.255.248.0	13	11	Range 4/7
	62	2^{10-2}	1,022	255.255.252.0	14	10	Range 5/7
	126	2^{9-2}	510	255.255.254.0	15	9	Range 6/7
Class C	0	2^{8-2}	254	255.255.255.0	16	8	B' Range 7/7 & Start C
	Invalid	2^{7-2}	Invalid	128 = invalid	17	7	
	2	2^{6-2}	62	255.255.255.192	18	6	Range 1/5
	6	2^{5-2}	30	255.255.255.224	19	5	Range 2/5
	14	2^{4-2}	14	255.255.255.240	20	4	Range 3/5
	30	2^{3-2}	6	255.255.255.248	21	3	Range 4/5
	62	2^{2-2}	2	255.255.255.252	22	2	Range 5/5

SubNets per Mask (Short-Form) - MEMORIZE

Class A	Bits Req (n)		Max Subs (2^n-2)		Hosts Per (2^(24-n)-2)		Masks (Decimal)	
Class A	2		4-2=2		4,194,302		255.192.0.0	Sub 1/7
	3		8-2=6		2,097,150		255.224.0.0	Sub 2/7
	4		16-2=14		1,048,574		255.240.0.0	Sub 3/7
	5		32-2=30		524,286		255.248.0.0	Sub 4/7
	6		64-2=62		262,142		255.252.0.0	Sub 5/7
	7 *		128-2=126		131,070		255.254.0.0	Not Class C
	8 *		256-2=254		65,534		255.255.0.0	Not Class C

Class B	Bits Req (n)		Max Subs (2^n-2)		Hosts Per (2^(16-n)-2)		Masks (Decimal)	
Class B	2		4-2=2		16,382		255.255.192.0	Sub 1/7
	3		8-2=6		8,190		255.255.224.0	Sub 2/7
	4		16-2=14		4,094		255.255.240.0	Sub 3/7
	5		32-2=30		2,046		255.255.248.0	Sub 4/7
	6		64-2=62		1,022		255.255.252.0	Sub 5/7
	7 *		128-2=126		510		255.255.254.0	Not Class C
	8 *		256-2=254		254		255.255.255.0	Not Class C

Class C	Bits Req (n)		Max Subs (2^n-2)		Hosts Per (2^(8-n)-2)		Masks (Decimal)	
Class C	2		4-2=2		62		255.255.255.192	Sub 1/5
	3		8-2=6		30		255.255.255.224	Sub 2/5
	4		16-2=14		14		255.255.255.240	Sub 3/5
	5		32-2=30		6		255.255.255.248	Sub 4/5
	6		64-2=62		2		255.255.255.252	Sub 5/5

Shortcuts:
1. There are 7 valid SubNet Masks for Classs A&B, and 5 valid for Class C.
2. To be valid, all SubNet Masks must use at least 2 bits, no matter what Class.
3. Based on (*n*) number of Network ID bits, (2^n-2) = Maximum number of SubNets per bit mask.
   Example: You are given 4 bits. (2^4-2) = (16-2) = 14 SubNets per 4-bit Mask. (Which is x.y.z.240).
4. Based on (*h*) number of Host bits, and (*n*) Network ID bits, (2^(h-n)-2) = Maximum # of Hosts per SubNet.
   Example: The Class of Example #3 above is "B"…
   ... (2^(16-4)-2) = (2^(12))-2 = 4096-2 = 4094 Hosts per 4-bit "B" SubNet.

- The 2-bit Class C breakdown -

Valid uses: 255.255.255.192 / 255.255.192.0 / 255.192.0.0

*Network 1/4: Invalid.			
	00/000000*	0*	Invalid
	00/000001*	1*	Invalid
	00/111110*	62*	Invalid
	00/111111*	63*	Invalid
Network 2/4:			
	01/000000	64	All 0's for host is invalid.
Valid 1/2	01/000001	65	Start of valid net 2 of 4.
	01/111110	126	End of valid net 2 of 4.
	01/111111	127	Broadcast for net 2 of 4.
Network 3/4:			
	10/000000	128	All 0's for host is invalid.
Valid 2/2	10/000001	129	Start of valid net 3 of 4.
	10/111110	190	End of valid net 3 of 4.
	10/111111	191	Broadcast for net 4 of 4.
*Network 4/4: Invalid.			
	11/000000*	192*	Invalid
	11/000001*	193*	Invalid
	11/111110*	254*	Invalid
	11/111111*	255*	Invalid

- The 3-bit Class C breakdown -

Valid uses: 255.255.255.224 / 255.255.224.0 / 255.224.0.0

*Network 1/8: Invalid			
	000/00000*	0*	Invalid
	000/00001*	1*	Invalid
	000/11110*	30*	Invalid
	000/11111*	31*	Invalid
Network 2/8:			
	001/00000	32	All 0's for host is invalid.
Valid 1/6	001/00001	33	Start of valid net 2 of 8.
	001/11110	62	End of valid net 2 of 8.
	001/11111	63	Broadcast for net 2 of 8.
Network 3/8:			
	010/00000	64	All 0's for host is invalid.
Valid 2/6	010/00001	65	Start of valid net 3 of 8.
	010/11110	94	End of valid net 3 of 8.
	010/11111	95	Broadcast for net 3 of 8.
Network 4/8:			
	011/00000	96	All 0's for host is invalid.
Valid 3/6	011/00001	97	Start of valid net 4 of 8.
	011/11110	126	End of valid net 4 of 8.
	011/11111	127	Broadcast for net 4 of 8.
Network 5/8:			
	100/00000	128	All 0's for host is invalid.
Valid 4/6	100/00001	129	Start of valid net 5 of 8.
	100/11110	158	End of valid net 5 of 8.
	100/11111	159	Broadcast for net 5 of 8.
Network 6/8:			
	101/00000	160	All 0's for host is invalid.
Valid 5/6	101/00001	161	Start of valid net 6 of 8.
	101/11110	190	End of valid net 6 of 8.
	101/11111	191	Broadcast for net 6 of 8.
Network 7/8:			
	110/00000	192	All 0's for host is invalid.
Valid 6/6	110/00001	193	Start of valid net 7 of 8.
	110/11110	222	End of valid net 7 of 8.
	110/11111	223	Broadcast for net 7 of 8.
*Network 8/8: Invalid			
	111/00000*	224*	Invalid
	111/00001*	225*	Invalid
	111/11110*	254*	Invalid
	111/11111*	255*	Invalid

Here are some shortcuts to memorize:

NOTE: For every octet going to the left, multiply by 256 for the next higher Class (A, B, C)

Example: Suppose Class C mask of 255.255.255.192 uses 2 bits
This gives you 4-2=2 subnets, and 64-2=62 hosts per subnet
So, to convert this mask to Class B and compute hosts per subnet,
…assuming you are using 255.255.192.0 subnet mask…..
and take ((64*256)-2) = (16,384-2) = 16,382 hosts per Class B .192 subnet
In other words, to upgrade hosts per subnet, simply multiply by 256 per Class, and subtract by 2.

NOTE: Only 7 subs valid for Class A or B. Class C has only 5 valid, as the last 2 are not used.

Example: Subnet masks with last octet of 254 or 255 are only valid for Classes A and B; not Class C.
Suppose you want to use Class B 255.255.254.0 subnet mask.
For Class B, this gives you 128-2=126 subnets, with (2^(16-7)-2)=510 hosts per subnet.
For Class C, this is would be 255.255.255.254, and would be invalid.

NOTE: Use net-side n bits: (2^*n* -2)=x to get total number of SubNets. (*n* =net bits).

Example: OK, now we are back to Class C (8 bits). You are allowed 3 Mask bits...
This means you can have ((2^3)-2) = (8-2) = 6 Subnets from this Mask.
All 3 bits (11100000) = 128+64+32 = 224 SubNet Mask (decimal).

NOTE: Use remaining host-side *h* bits: (2^(h-n)-2)=x to get total number of Hosts per SubNet.

Example: Let's say you are still using those 3 bits for Class C Mask above…….
You have (8-3)=5 bits left for hosts, assuming Class C. (*n* =net bits, *h* =host bits).
This means you can have (2^(8-5)-2) = (32-2) = 30 Hosts per Subnet, INCLUDING network ID.
All remaining bits (00011111) = 16+8+4+2+1= (31-2) = 29 useable Hosts per Subnet

NOTE: To upgrade Class, simply multiple by 256 and subtract 2.

Example: Let's say you just figured all of this out and assumed wrong, it's really a Class B.
You STILL have only 6 Subnets from this mask, but suddenly have 8 more bits for the Host side.
All you are doing here is shifting all octets over to the left by one. No big deal.
Simply take the ORIGINAL Host bits (2^5) = 32 Hosts per Subnet, multiply by 256, and subtract 2.
This means that you now have ((32 * 256) - 2) = (512-2) = 510 Hosts per SubNet

NOTE: The number of Hosts, when based on the Mask, is always based double the previous number.

Example: When you take (2^2)=4 hosts, the next higher is always (2^3)=8 hosts, which is double
Thus, you always subtract 2 from this number, and get the number of Hosts per Mask.
Warning: Speed is the name of the game for this exam!

Here are some super-simple charts to memorize:

Memorize: **"Bit combinations per Mask"**

Decimal	Hosts per	Class Valid
192 Mask	(64-2)=62	Any Class 1/7
224 Mask	(32-2)=30	Any Class 2/7
240 Mask	(16-2)=14	Any Class 3/7
248 Mask	(8-2)= 6	Any Class 4/7
252 Mask	(4-2)= 2	Any Class 5/7
254 Mask *	(2-1)=1	Class A or B only
255 Mask *	(1-1)=0	Class A or B only

Quiz yourself 1/2: "What is the Mask for xx Hosts in Class C?"
Quiz yourself 2/2: "How many Hosts for xx Mask in Class C?"

Memorize: **"Valid SubNet Masks"**

11000000	2 bits	192 mask	Any Class
11100000	3 bits	224 mask	Any Class
11110000	4 bits	240 mask	Any Class
11111000	5 bits	248 mask	Any Class
11111100	6 bits	252 mask	Any Class
11111110 *	7 bits	254 mask	Class A or B only
11111111 *	8 bits	255 mask	Class A or B only

Quiz yourself: "What are the 7 valid subnet mask bits?"

Memorize: **"Valid Subnets per Bit"**

Mask bits	Delta	Subnets per
2 bits	64	2 subnets
3 bits	32	6 subnets
4 bits	16	14 subnets
5 bits	8	30 subnets
6 bits	4	62 subnets
7 bits	2	126 subnets
8 bits	1	254 subnets

Quiz yourself: "How many SubNets are in those 7 Masks?"

Memorize: **"Delta of each Mask"**

192 mask	Delta = 64	Any Class
224 mask	Delta = 32	Any Class
240 mask	Delta = 16	Any Class
248 mask	Delta = 8	Any Class
252 mask	Delta = 4	Any Class
254 mask	Delta = 2	Class A or B only
255 mask	Delta = 1	Class A or B only

Quiz yourself: "What is the starting Delta number of each of the 7 Masks?"

Memorize: **"Octal Bit Breakdown"**

Binary 128 + 64 + 32 + 16 + 8 + 4 + 2 + 1 = Decimal 255

Quiz yourself 1/2: "What is 255 in binary?"
Quiz yourself 2/2: "What are all 8 binary bits in Decimal?"

Memorize: **"Valid SubNets per Mask"**

192 mask	2 subnets	Any Class
224 mask	6 subnets	Any Class
240 mask	14 subnets	Any Class
248 mask	30 subnets	Any Class
252 mask	62 subnets	Any Class
254 mask *	126 subnets	Class A or B only
255 mask *	254 subnets	Class A or B only

Quiz yourself: "How many Subnets do you get on which masks?"

Memorize: **"Address Class Ranges"**

Class A	1-126	Network ID 0
Class B	128-191	Network ID 10
Class C	192-223	Network ID 110

Quiz yourself 1/3: "What is the range for Class A?"
Quiz yourself 2/3: "What is the range for Class B?"
Quiz yourself 3/3: "What is the range for Class C?"

Super-Supreme Final-Exam Memorization Chart

SubNet Mask	Subnet Bits	Super Delta	Subs Per Mask	Hosts per SubNet Class A - by Delta	by Bit Class A	Total Hosts per SubNet Class A
0	0	256	(1-1)=0	(256*256*256)-2	2^(24-n)-2	16,777,214
128	1	128	(2-1)=1	invalid	invalid	invalid
192	2	64	(4-2)=2	(256*256*64)-2	2^(24-n)-2	4,194,302
224	3	32	(8-2)=6	(256*256*32)-2	2^(24-n)-2	2,097,150
240	4	16	(16-2)=14	(256*256*16)-2	2^(24-n)-2	1,048,574
248	5	8	(32-2)=30	(256*256*8)-2	2^(24-n)-2	524,286
252	6	4	(64-2)=62	(256*256*4)-2	2^(24-n)-2	262,142
254	7	2	(128-2)=126	(256*256*2)-2	2^(24-n)-2	131,070
255	8	1	(256-2)=254	(256*256*1)-2	2^(24-n)-2	65,534

SubNet Mask	Subnet Bits	Super Delta	Subs Per Mask	Hosts per SubNet Class B - by Delta	by Bit Class B	Total Hosts per SubNet Class B
0	0	256	(1-1)=0	(256*256)-2	2^(16-n)-2	65,534
128	1	128	(2-1)=1	invalid	invalid	invalid
192	2	64	(4-2)=2	(256*64)-2	2^(16-n)-2	16,382
224	3	32	(8-2)=6	(256*32)-2	2^(16-n)-2	8,190
240	4	16	(16-2)=14	(256*16)-2	2^(16-n)-2	4,094
248	5	8	(32-2)=30	(256*8)-2	2^(16-n)-2	2,046
252	6	4	(64-2)=62	(256*4)-2	2^(16-n)-2	1,022
254	7	2	(128-2)=126	(256*2)-2	2^(16-n)-2	510
255	8	1	(256-2)=254	(256*1)-2	2^(16-n)-2	254

SubNet Mask	Subnet Bits	Super Delta	Subs Per Mask	Hosts per SubNet Class C - by Delta	by Bit Class C	Total Hosts per SubNet Class C
0	0	256	(1-1)=0	(256)-2	2^(8-n)-2	254
128	1	128	(2-1)=1	invalid	invalid	invalid
192	2	64	(4-2)=2	(64)-2	2^(8-n)-2	62
224	3	32	(8-2)=6	(32)-2	2^(8-n)-2	30
240	4	16	(16-2)=14	(16)-2	2^(8-n)-2	14
248	5	8	(32-2)=30	(8)-2	2^(8-n)-2	6
252	6	4	(64-2)=62	(4)-2	2^(8-n)-2	2

This one is probably the most important to memorize, so you can get Delta # per Subnet.
Example "Big-Number" question. Have Class A mask of 255.224.0.0. How many Hosts per SubNet?

FIRST, I'll work it out by Delta……..

1. There are 3 bits per 224 mask no matter what. This is ALWAYS based on 32 Delta.
2. Shift the octet over to the left by Two (8+8=16 bits), Multiply by 256 each 8-bit octet, then Subtract 2.
3. So, using 3 bits, take ((32 * 256 * 256)-2) = 2,097,150 Hosts per "Class A" 224 SubNet.

SECOND, I'll work it out by Powers-of-Two……..

1. There are 3 bits per 224 mask no matter what. Memorize this or fail the exam.
2. Since this is Class "A", there are 8+8+8=24 Host Bits involved…
3. Where (*h*) is number of Host Bits (24), and,
4. Where (*n*) is number of NetMask Bits (3), so…
5. Using 3 bits, take (2^(h-n)-2) = (2^(24-3)-2) = (2,097,152)-2 = 2,097,150 Hosts per "Class A" 224 SubNet.

An easy way to see how the "(2^(HostBits-NetMaskBits)-2) = Hosts Per SubNet" equation works:

Working out a full Class C, there are 8 bits in the Host range.
And, there are 0 bits being robbed by the Mask range.
This means we're talking about a standard subnet mask of 255.255.255.0.
So, the equation is (2^(h-n)-2) == (2^(8-0)-2) == (2^(8)-2) == (2^8)-2 == (256-2) == 254
Bingo! There are 254 useable hosts in a Class C with standard mask of 255.255.255.0

An easy way to see how the "Delta" shortcut works. This is up to you, but I think it's faster….

Working out a full Class C, there are 8 bits in the Host range.
And, there are 0 bits being robbed by the Mask range.
This means we're talking about a standard subnet mask of 255.255.255.0.
The Delta is 256 for subs per mask of 0 == (256-2) == 254.
Since you've memorized the Delta chart, you knew that "0" bits = "256" Delta.
Bingo! Now, wasn't that easier?