Valid host range?

for **158 . 167 . 18 . 156 / 15**

1) Translate to binary each 8 bit section

158 . 167 . 18 . 156

10011110 . 1010011[1] . 00010010 . 10011100

2) Based on the CIDR, look for 15 th bit

3) Starting on the next bit, set:
   - all the bits to 0

10011110 . 101001[1]0 . 00000000 . 00000000

   - and translate the result to decimal, this will give you the <u>network address</u>

156 . 166 . 0 . 0

4) Same as 3), but after the CIDR bit
   - set all bits to 1

10011110 . 101001[1]1 . 11111111 . 11111111

   - translate the result back to decimal, this will give you the <u>broadcast address</u>

156 . 167 . 255 . 255

5) The needed range = all between the network address and the brodcast address, with them __NOT__ included:

156.166.0.1 $\longrightarrow$ 156.167.255.254

For questions of type "which host address could communicate with this IP" $\longrightarrow$ __any host within this range__

Max number of IP addresses?

A) Based on IPv4 with CIDR

- the IP format gives us the number of bits used, we don't care about its content, we only need <u>bits</u> and CIDR

$$124.144.156.248 / \boxed{21}$$

32 bits

Method: $2^{(32 - \boxed{21})} = 2^{11} = 2048$

From the result, substract 2 (network and broadcast address don't count)

$$2048 - 2 = \boxed{2046} \leftarrow \text{Final answer}$$

B) Based on a subnet mask

255.224.0.0

1) Translate to binary

255 . 224 . 0 . 0

11111111 . 11100000 . 00000000 . 00000000

2) Count the number of "1" → **11**

This gives you the CIDR notation

3) As seen in the A example, just do:

$$2^{32-\textcolor{red}{11}} = 2^{21} = 2097152 - 2 = \boxed{2097150}$$

<u>Note</u>: in some questions, different IP type can be used. In such case, replace 32 with the number of bits used

<u>Example</u>:

255 . 255 . 255 . 255 . 128

8 + 8 + 8 + 8 + 8 = $\boxed{40}$ bits

128 in binary → $\boxed{1}$ 0000000
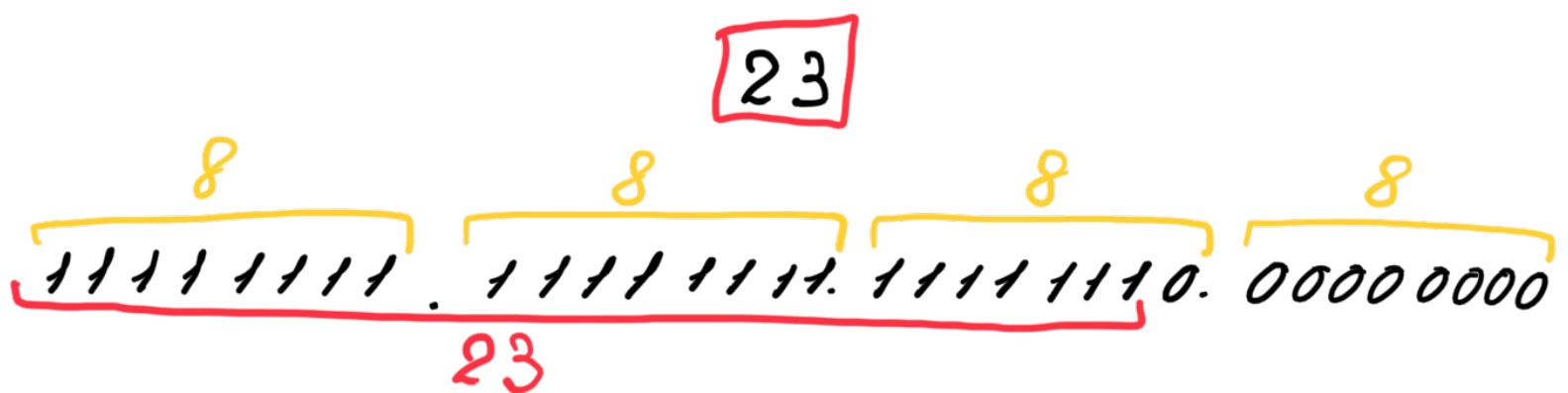
↳ **33rd bit**

$$2^{\textcolor{orange}{40}-\textcolor{red}{33}} = 2^7 = 128 - 2 = \boxed{126}$$

Note 2:

To find a subnet mask based on CIDR

Example: CIDR = /23

1) CIDR gives you the number of 1
in the binary notation of your mask.
All the rest is set to 0.

[23]

| 8 | 8 | 8 | 8 |

1111 1111 . 1111 1111. 1111 1110. 0000 0000

23

2) Just translate the result to decimal

1111 1111 . 1111 1111. 1111 1110. 0000 0000

255 . 255 . 254 . 0