Interesting Doubt → ASCII CODE

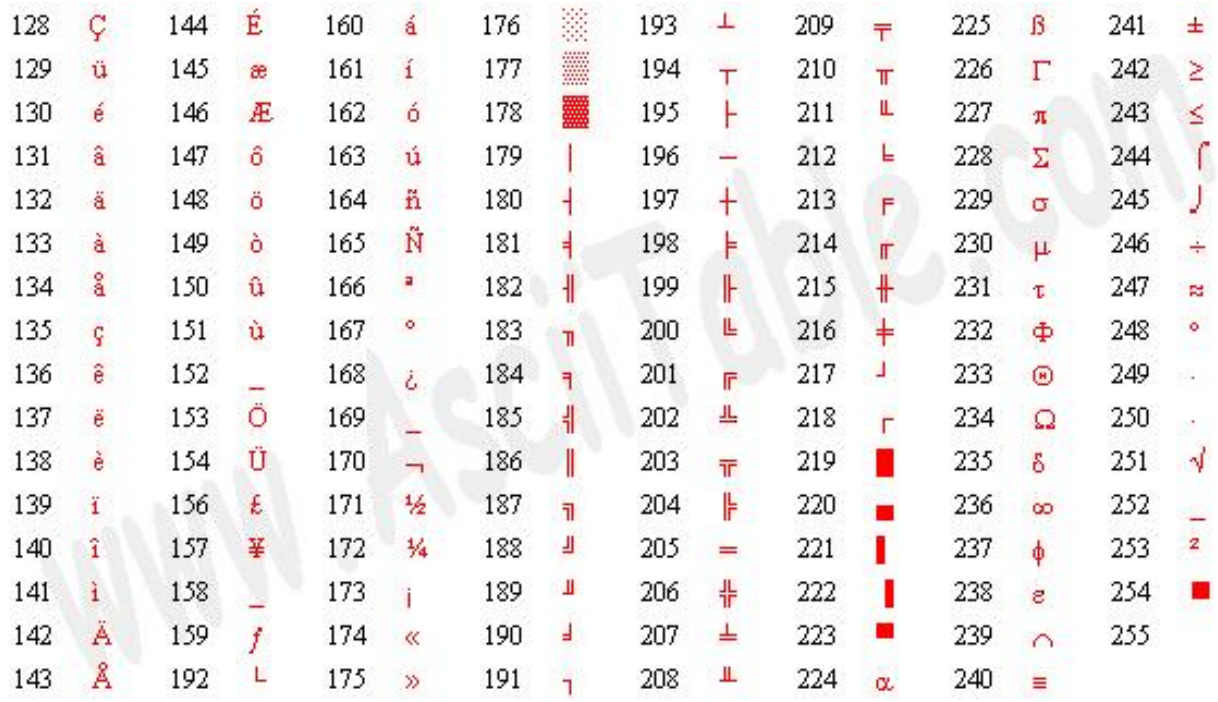
If we talk about unsigned char, then the value range is 0 to 255 and If we talk about signed char then the range is -128 to 127.

The Question is what does value from [-128 to 0] represent and are they same as values in range [128 to 255] in case of unsigned chars?

Solution: We can clearly see the value to character using the below links.

For Range 0 to 127: Check this Out

for Range 128 to 255:



128	Ç	144	É	160	á	176	▒	193	┴	209	╤	225	ß	241	±
129	ü	145	æ	161	í	177	▒	194	┬	210	╥	226	Γ	242	≥
130	é	146	Æ	162	ó	178	▓	195	├	211	╙	227	π	243	≤
131	â	147	ô	163	ú	179	│	196	─	212	╘	228	Σ	244	⌠
132	ä	148	ö	164	ñ	180	┤	197	┼	213	╒	229	σ	245	⌡
133	à	149	ò	165	Ñ	181	╡	198	╞	214	╓	230	µ	246	÷
134	å	150	û	166	ª	182	╢	199	╟	215	╫	231	τ	247	≈
135	ç	151	ù	167	º	183	╖	200	╚	216	╪	232	Φ	248	°
136	ê	152	_	168	¿	184	╕	201	╔	217	┘	233	⊕	249	·
137	ë	153	Ö	169	_	185	╣	202	╩	218	┌	234	Ω	250	·
138	è	154	Ü	170	¬	186	║	203	╦	219	▉	235	δ	251	√
139	ï	156	£	171	½	187	╗	204	╠	220	▄	236	∞	252	_
140	î	157	¥	172	¼	188	╝	205	═	221	▌	237	ϕ	253	²
141	ì	158	_	173	¡	189	╜	206	╬	222	▐	238	ε	254	■
142	Ä	159	ƒ	174	«	190	╛	207	╧	223	▀	239	∩	255	
143	Å	192	└	175	»	191	┐	208	╨	224	α	240	≡		

This ASCII characters are same for range [-128 to -1]

It is clearly visible that the range 0 to 127 contains normal characters in both the signed and unsigned cases but [-128 to -1] range and [128 to 255] range contains “extended characters” as visible in the list attached above.

here, we can easily comprehend that:

128 is exactly equal to -128 in Binary representation and mapping to the same character.

129 is exactly equal to -127 in Binary representation and mapping to the same character.

130 is exactly equal to -126 in Binary representation and mapping to the same character and so on.

In most of the machines, when we try to print the characters in the IDE/Code_Editor, we get diamond shaped question mark symbol is basically,

“The replacement character , a black diamond with a white question mark, is a symbol found in the Unicode standard at code point U+FFFD in the Specials table. It is used to replace an unknown or unrepresentable character, or indicate problems when a system is not able to render a stream of data to a correct symbol.”