

Table of Contents

ASCII Tablosu ve C++'ta Karakter Temsili	3
<i>Karakterin ASCII Deęerini Bulma</i>	4
<i>Büyük Harf / Küçük Harf Dönüşümü (Karakter Aritmetiđi)</i>	4

ASCII Tablosu ve C++'ta Karakter Temsili

Snippet from [Wikipedia: ASCII](#)

ASCII (İngilizce: **American Standard Code for Information Interchange**, Türkçe: **Bilgi Değişimi İçin Amerikan Standart Kodlama Sistemi**) Latin alfabesi üzerine kurulu 7 bitlik bir karakter kümesidir. İlk kez 1963 yılında ANSI tarafından standart olarak sunulmuştur.

ASCII'de 33 tane basılmayan kontrol karakteri ve 95 tane basılan karakter bulunur. Kontrol karakterleri metnin akışını kontrol eden, ekranda çıkmayan karakterlerdir. Basılan karakterler ise ekranda görünen, okuduğumuz metni oluşturan karakterlerdir. ASCII'nin basılan karakterleri aşağıda belirtilmiştir.

[Creative Commons Attribution-Share Alike 4.0](#)

C++ programlamada karakterler bellekte doğrudan metin olarak değil, sayısal değerler olarak tutulur. Bu sayısal değerlerin karakterlerle nasıl eşleştiğini belirleyen evrensel standarda ASCII¹⁾ denir.

C++'da temel karakter veri tipi olan `char`, özünde 8 bitlik bir tam sayı tipidir. Genellikle arka planda ASCII tablosunu kullanır.

Bir karaktere işlem yaptığımızda aslında onun ASCII değerine işlem yaparız. Bu durum bize *karakter aritmetiği* dediğimiz güçlü ve hızlı bir manipülasyon tekniklerinin kapısını açar.

Code	Symbol	Code	Symbol	Code	Symbol	Code	Symbol
0	NUL (null)	32	(space)	64	@	96	`
1	SOH (start of header)	33	!	65	A	97	a
2	STX (start of text)	34	"	66	B	98	b
3	ETX (end of text)	35	#	67	C	99	c
4	EOT (end of transmission)	36	\$	68	D	100	d
5	ENQ (enquiry)	37	%	69	E	101	e
6	ACK (acknowledge)	38	&	70	F	102	f
7	BEL (bell)	39	'	71	G	103	g
8	BS (backspace)	40	(72	H	104	h
9	HT (horizontal tab)	41)	73	I	105	i
10	LF (line feed/new line)	42	*	74	J	106	j
11	VT (vertical tab)	43	+	75	K	107	k
12	FF (form feed / new page)	44	,	76	L	108	l
13	CR (carriage return)	45	-	77	M	109	m
14	SO (shift out)	46	.	78	N	110	n
15	SI (shift in)	47	/	79	O	111	o
16	DLE (data link escape)	48	0	80	P	112	p
17	DC1 (data control 1)	49	1	81	Q	113	q
18	DC2 (data control 2)	50	2	82	R	114	r
19	DC3 (data control 3)	51	3	83	S	115	s
20	DC4 (data control 4)	52	4	84	T	116	t

Code	Symbol	Code	Symbol	Code	Symbol	Code	Symbol
21	NAK (negative acknowledge)	53	5	85	U	117	u
22	SYN (synchronous idle)	54	6	86	V	118	v
23	ETB (end of transmission block)	55	7	87	W	119	w
24	CAN (cancel)	56	8	88	X	120	x
25	EM (end of medium)	57	9	89	Y	121	y
26	SUB (substitute)	58	:	90	Z	122	z
27	ESC (escape)	59	;	91	[123	{
28	FS (file separator)	60	<	92	\	124	
29	GS (group separator)	61	=	93]	125	}
30	RS (record separator)	62	>	94	^	126	~
31	US (unit separator)	63	?	95	_	127	DEL (delete)

Karakterin ASCII Deęerini Bulma

```
#include <iostream>

int main(){
char harf = 'A';

// Güvenli ve modern C++ dönüşümü (Casting)
int asciiDegeri = static_cast<int>(harf);

std::cout << "Karakter: " << harf << "\n";
std::cout << "ASCII Degeri: " << asciiDegeri << "\n"; // Çıktı: 65

return 0;
}
```

Büyük Harf / Küçük Harf Dönüşümü (Karakter Aritmetięi)

Büyük harfler ile küçük harfler arasında ASCII tablosunda tam olarak 32 birim fark vardır. (Örn: 'a' = 97, 'A' = 65 → 97 - 65 = 32). Bu kuralı kullanarak manuel büyük/küçük harf dönüşümü yapabiliriz.

```
#include <iostream>

int main() {
char kucukHarf = 'g';

// Küçük harften 32 çıkarırsak büyük harfi elde ederiz.
char buyukHarf = kucukHarf - 32;

std::cout << kucukHarf << " harfinin buyugu: " << buyukHarf << "\n";
// Çıktı: g harfinin buyugu: G
}
```

```
return 0;  
  
}
```

Not: Modern C++ projelerinde bu işlemi manuel yapmak yerine `<cctype>` kütüphanesindeki `std::toupper()` ve `std::tolower()` fonksiyonlarını kullanmak en iyi pratiktir

Bu belgedeki bilgiler [UCH Wiki](#)'den alınmıştır.

1)

American Standard Code for Information Interchange

From:

<https://wiki.ulascemh.com/> - **UCH**

Permanent link:

<https://wiki.ulascemh.com/doku.php?id=tr:cs:lang:c++:types:ascii&rev=1776000666>

Last update: **2026/04/12 13:31**

