Lecture #4: ASCII System in C Language

ASCII System

Stands for "American Standard Code for Information Interchange." ASCII is a character encoding that uses numeric codes to represent characters. These include upper and lowercase English letters, numbers, and punctuation symbols.

ASCII
American **S**tandard **C**ode for **I**nformation **I**nterchange

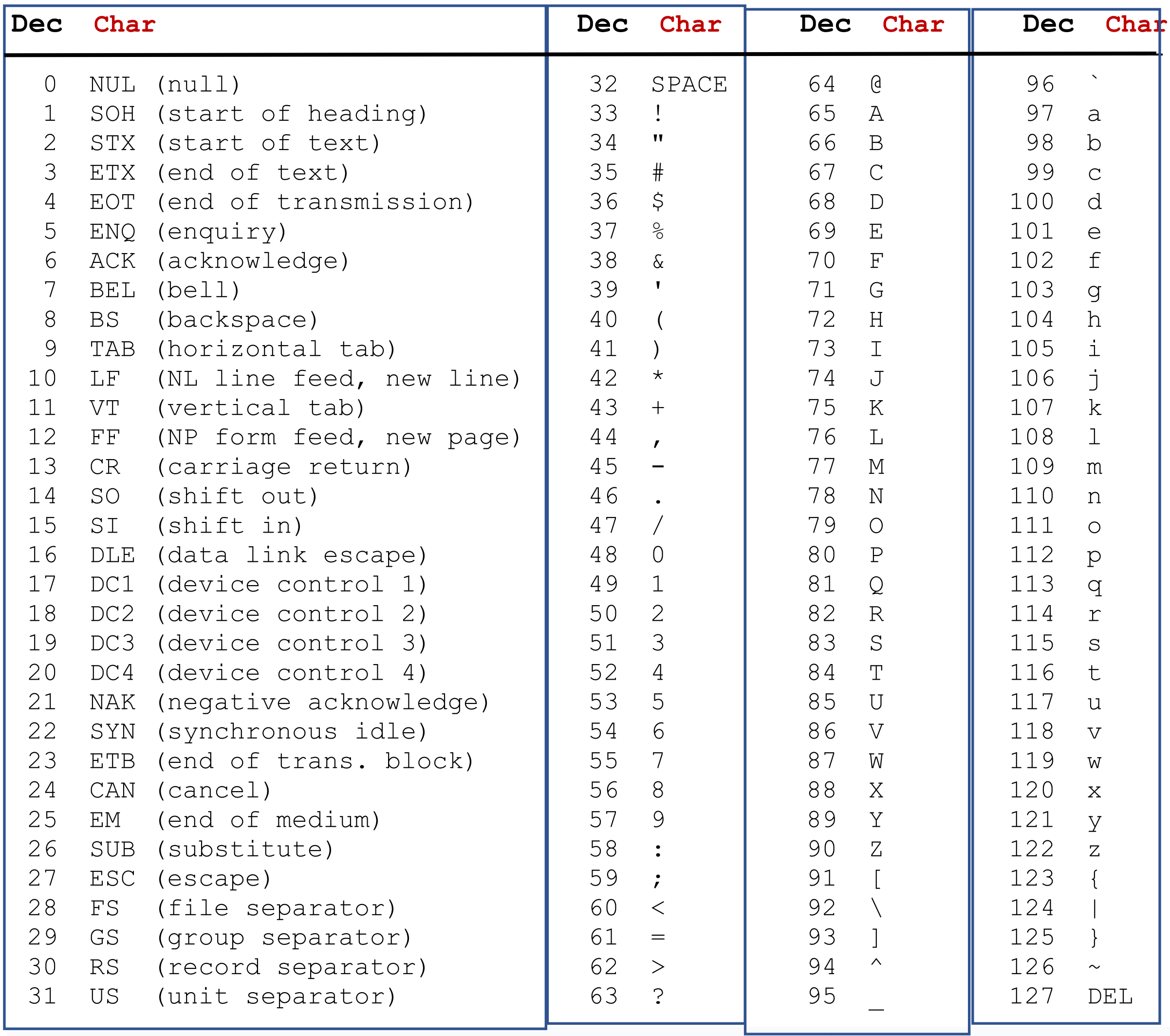
Mean :

- As the name suggest "**code for Info interchange**". Well it do the same thing.
- It is used to interchange information.
- Computer doesn't understand English or Hindi or any language.
- But computer do understand only **0** and **1**.
- **0** means **False**
- **1** means **True**
- **ASCII codes** represent text in computers, communications equipment, and other devices that use text.
- If you press **4** from keyboard, then keyboard send the value **100**(value equivalent to 4) to the **main memory** which will further be evaluated in **secondary memory** and then the output will be 4 in monitor.
- Note that **100** means true-false-false.
- Similarly if you press "**a"** the value will be **097** which is equivalent to **01100001.**

ASCII Table

Dec = Decimal Value
Char = Character

'5' has the int value 53
if we write '5'-'0' it evaluates to 53-48, or the int 5
if we write char c = 'B'+32; then c stores 'b'



Dec	Char	Dec	Char	Dec	Char	Dec	Char
0	NUL (null)	32	SPACE	64	@	96	`
1	SOH (start of heading)	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	TAB (horizontal tab)	41	)	73	I	105	i
10	LF (NL line feed, new line)	42	*	74	J	106	j
11	VT (vertical tab)	43	+	75	K	107	k
12	FF (NP 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 (device control 1)	49	1	81	Q	113	q
18	DC2 (device control 2)	50	2	82	R	114	r
19	DC3 (device control 3)	51	3	83	S	115	s
20	DC4 (device control 4)	52	4	84	T	116	t
21	NAK (negative acknowledge)	53	5	85	U	117	u
22	SYN (synchronous idle)	54	6	86	V	118	v
23	ETB (end of trans. 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

In C programming, a character variable holds ASCII value **(an integer number between 0 and 127)** rather than that character itself. This integer value is the ASCII code of the character.

For example, the ASCII value of `'A'` is 65.

What this means is that, if you assign `'A'` to a character variable, 65 is stored in the variable rather than `'A'` itself.

Now, let's see how we can print the ASCII value of characters in C programming.

Program to Print ASCII Value

```c
#include <stdio.h>
int main() {
    char c;
    printf("Enter a character: ");
    scanf("%c", &c);

    // %d displays the integer value of a character
    // %c displays the actual character
    printf("ASCII value of %c = %d", c, c);

    return 0;
}
```

Output

```
Enter a character: G
ASCII value of G = 71
```

In this program, the user is asked to enter a character. The character is stored in variable `c`. When `%d` format string is used, **71** (the ASCII value of `G`) is displayed. When `%c` format string is used, `'G'` itself is displayed.

Source: Techterms.com || CodeCogs Contributors.

CodeCogs