

# 문자형과 아스키코드

## scanf\_s 와 버퍼

`scanf_s` 함수는 표준 입력으로부터 데이터를 읽어들이는데, 사용자가 입력한 데이터는 먼저 입력 버퍼에 저장됩니다. 사용자가 엔터를 치면, 그 입력(문자와 개행 문자 '\n')이 버퍼에 들어가고 `scanf_s` 는 버퍼에서 데이터를 가져옵니다.

## getchar() 의 사용

이 문제를 해결하기 위해 `getchar()` 함수를 사용합니다. `getchar()` 는 버퍼에서 한 문자를 읽고 제거하는 기능을 합니다. 첫 번째 문자를 입력받은 후 `getchar()` 를 호출하여 남아 있는 개행 문자를 제거함으로써, 두 번째 `scanf_s` 가 올바른 문자를 읽을 수 있도록 합니다.

## 예시 코드

```
#include <stdio.h>
int main() {
    char ch1, ch2;

    printf("첫 번째 문자를 입력하세요: ");
    scanf_s("%c", &ch1, 1);
    getchar(); // 첫 번째 입력 후 남은 개행 문자 제거

    printf("두 번째 문자를 입력하세요: ");
    scanf_s("%c", &ch2, 1);

    printf("입력된 문자: %c, %c\n", ch1, ch2);

    return 0;
}
```

이 코드에서 첫 번째 `scanf_s` 후에 `getchar()` 를 호출하여 개행 문자를 제거합니다. 이렇게 하면 두 번째 `scanf_s` 가 정상적으로 두 번째 문자 입력을 받을 수 있습니다.

## 아스키 코드 (ASCII Code)



아스키코드(ASCII 코드)는 컴퓨터가 글자를 이해하고 표현하는 방법 중 하나예요. 컴퓨터는 우리가 쓰는 일반 글자를 바로 이해할 수 없어요. 그래서 컴퓨터는 각 글자를 숫자로 바꿔서 기억해요. 예를 들어, 컴퓨터는 'A'라는 글자를 보고 '65'라는 숫자로 생각해요. 마찬가지로, 'B'는 '66', 'C'는 '67'처럼 말이죠.

이렇게 컴퓨터가 글자를 숫자로 바꾸는 규칙을 '아스키코드'라고 해요. 이 규칙 덕분에 우리는 컴퓨터에 글자를 쓸 수 있고, 컴퓨터도 우리가 쓴 글자를 읽을 수 있어요.

제어 문자		공백 문자		구두점		숫자		알파벳			
10진	16진	문자	10진	16진	문자	10진	16진	문자	10진	16진	문자
0	0x00	NUL	32	0x20	SP	64	0x40	@	96	0x60	`
1	0x01	SOH	33	0x21	!	65	0x41	A	97	0x61	a
2	0x02	STX	34	0x22	"	66	0x42	B	98	0x62	b
3	0x03	ETX	35	0x23	#	67	0x43	C	99	0x63	c
4	0x04	EOT	36	0x24	\$	68	0x44	D	100	0x64	d
5	0x05	ENQ	37	0x25	%	69	0x45	E	101	0x65	e
6	0x06	ACK	38	0x26	&	70	0x46	F	102	0x66	f
7	0x07	BEL	39	0x27	'	71	0x47	G	103	0x67	g
8	0x08	BS	40	0x28	(	72	0x48	H	104	0x68	h
9	0x09	HT	41	0x29	)	73	0x49	I	105	0x69	i
10	0x0A	LF	42	0x2A	*	74	0x4A	J	106	0x6A	j
11	0x0B	VT	43	0x2B	+	75	0x4B	K	107	0x6B	k
12	0x0C	FF	44	0x2C	,	76	0x4C	L	108	0x6C	l
13	0x0D	CR	45	0x2D	-	77	0x4D	M	109	0x6D	m
14	0x0E	SO	46	0x2E	.	78	0x4E	N	110	0x6E	n
15	0x0F	SI	47	0x2F	/	79	0x4F	O	111	0x6F	o
16	0x10	DLE	48	0x30	0	80	0x50	P	112	0x70	p
17	0x11	DC1	49	0x31	1	81	0x51	Q	113	0x71	q
18	0x12	DC2	50	0x32	2	82	0x52	R	114	0x72	r
19	0x13	DC3	51	0x33	3	83	0x53	S	115	0x73	s
20	0x14	DC4	52	0x34	4	84	0x54	T	116	0x74	t
21	0x15	NAK	53	0x35	5	85	0x55	U	117	0x75	u
22	0x16	SYN	54	0x36	6	86	0x56	V	118	0x76	v
23	0x17	ETB	55	0x37	7	87	0x57	W	119	0x77	w
24	0x18	CAN	56	0x38	8	88	0x58	X	120	0x78	x
25	0x19	EM	57	0x39	9	89	0x59	Y	121	0x79	y
26	0x1A	SUB	58	0x3A	:	90	0x5A	Z	122	0x7A	z
27	0x1B	ESC	59	0x3B	;	91	0x5B	[	123	0x7B	{
28	0x1C	FS	60	0x3C	<	92	0x5C	₩	124	0x7C	
29	0x1D	GS	61	0x3D	=	93	0x5D	]	125	0x7D	}
30	0x1E	RS	62	0x3E	>	94	0x5E	^	126	0x7E	~
31	0x1F	US	63	0x3F	?	95	0x5F	_	127	0x7F	DEL

## 예시: char 사용하기

C언어에서 char 타입을 사용하여 문자를 저장하고 출력해보겠습니다.

```
#include <stdio.h>
int main() {
    char letter = 'A'; // 'A'라는 문자를 저장
    printf("문자: %c\n", letter); // 문자 출력
    printf("아스키 코드: %d\n", letter); // 'A'의 아스키 코드인 65
출력

    return 0;
}
```

이 코드에서 `char letter = 'A';` 는 'A'라는 문자를 변수 letter에 저장합니다. `%c` 를 사용하여 문자를, `%d` 를 사용하여 해당 문자의 아스키 코드를 출력합니다.