C언어 강의자료

문정욱

프로그래밍 연습 6
표준 문자/문자열 처리 함수

문자 입출력 함수

EOF 체크

```c
#include <stdio.h>

int main(void)
{
    int ch;

    ch=getchar();
    putchar(ch);
    return 0;
}
```

```c
#include <stdio.h>

int main(void)
{
    int ch;

    ch=getchar();
    if( ch!=EOF )
        putchar(ch);
    return 0;
}
```

문자 입출력 함수

간소화

```c
#include <stdio.h>

int main(void)
{
    int ch;

    ch=getchar();
    if( ch!=EOF )
        putchar(ch);
    return 0;
}
```

```c
#include <stdio.h>

int main(void)
{
    int ch;

    if( (ch=getchar())!=EOF )
        putchar(ch);
    return 0;
}
```

문자 입출력 함수

두 번 반복 입력

```c
#include <stdio.h>

int main(void)
{
    int ch;

    if( (ch=getchar())!=EOF ) {
        putchar(ch);
    }
    return 0;
}
```

```c
#include <stdio.h>

int main(void)
{
    int ch;

    if( (ch=getchar())!=EOF ) {
        putchar(ch);
        if( (ch=getchar())!=EOF ) {
            putchar(ch);
        }
    }
    return 0;
}
```

문자 입출력 함수

세 번 반복 입력

```c
#include <stdio.h>

int main(void)
{
    int ch;

    if( (ch=getchar())!=EOF ) {
        putchar(ch);
        if( (ch=getchar())!=EOF ) {
            putchar(ch);
        }
    }
    return 0;
}
```

```c
#include <stdio.h>

int main(void)
{
    int ch;

    if( (ch=getchar())!=EOF ) {
        putchar(ch);
        if( (ch=getchar())!=EOF ) {
            putchar(ch);
            if( (ch=getchar())!=EOF ) {
                putchar(ch);
            }
        }
    }
    return 0;
}
```

문자 입출력 함수

여러 번 반복 입력

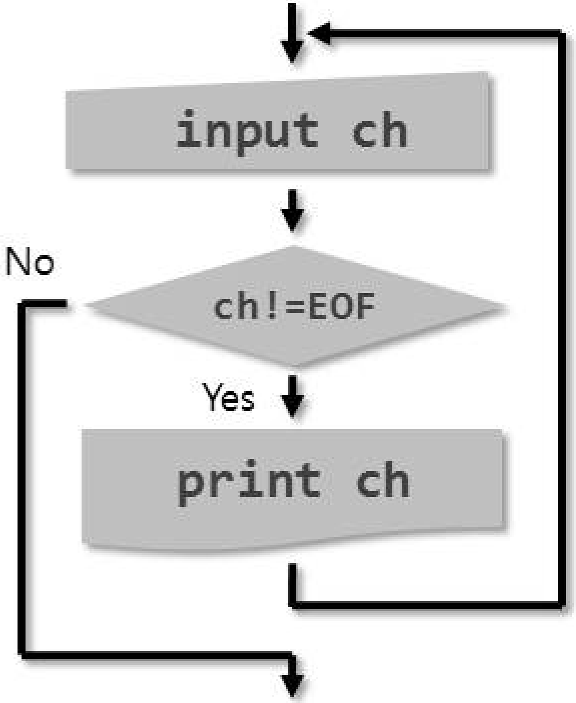
```
#include <stdio.h>

int main(void)
{
    int ch;

    if( (ch=getchar())!=EOF ) {
        putchar(ch);
        if( (ch=getchar())!=EOF ) {
            putchar(ch);
            if( (ch=getchar())!=EOF ) {
                putchar(ch);
            }
        }
    }
    return 0;
}
```

```
#include <stdio.h>

int main(void)
{
    int ch;

    if( (ch=getchar())!=EOF ) {
        putchar(ch);
        if( (ch=getchar())!=EOF ) {
            putchar(ch);
            if( (ch=getchar())!=EOF ) {
                putchar(ch);
                // repeat
            }
        }
    }
    return 0;
}
```

문자 입출력 함수

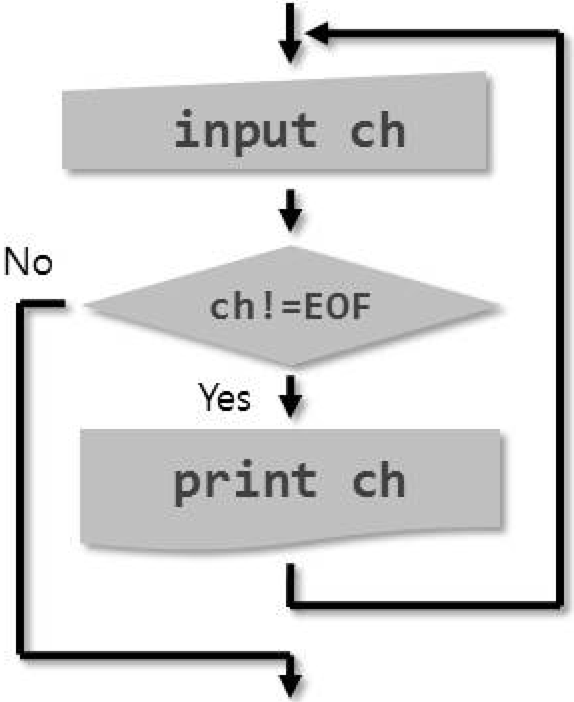
반복 입력의 순서도 표현

```c
#include <stdio.h>

int main(void)
{
    int ch;

    if( (ch=getchar())!=EOF ) {
        putchar(ch);
        if( (ch=getchar())!=EOF ) {
            putchar(ch);
            if( (ch=getchar())!=EOF ) {
                putchar(ch);
                // repeat
            }
        }
    }
    return 0;
}
```



문자 입출력 함수

순서도의 재해석



```
#include <stdio.h>

int main(void)
{
    int ch;

    while( (ch=getchar())!=EOF ) {
        putchar(ch);
    }
    return 0;
}
```

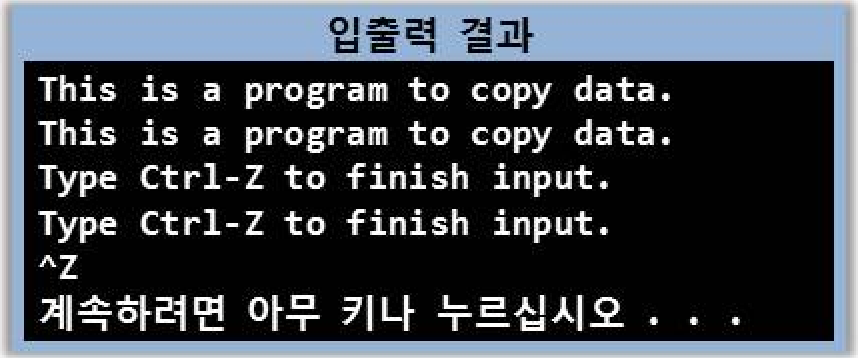
문자 입출력 함수

반복 입력의 간소화

```
#include <stdio.h>

int main(void)
{
    int ch;

    if( (ch=getchar())!=EOF ) {
        putchar(ch);
        if( (ch=getchar())!=EOF ) {
            putchar(ch);
            if( (ch=getchar())!=EOF ) {
                putchar(ch);
                // repeat
            }
        }
    }
    return 0;
}
```

```
#include <stdio.h>

int main(void)
{
    int ch;

    while( (ch=getchar())!=EOF ) {
        putchar(ch);
    }
    return 0;
}
```

```
c=getchar();
c!=EOF
```



입출력 결과

```
This is a program to copy data.
This is a program to copy data.
Type Ctrl-Z to finish input.
Type Ctrl-Z to finish input.
^Z
계속하려면 아무 키나 누르십시오 . . .
```

문자열 입출력 함수 만들기

```c
#include <stdio.h>

int my_gets(char buffer[])
{
    int i, ch;

    for(i=0;i<511 &&
        !((ch=getchar())==EOF || ch=='\n');++i)
        buffer[i] = ch;
    buffer[i]='\0';
    return i;
}

void my_puts(char buffer[])
{
    int i;

    for(i=0;buffer[i]!='\0'; ++i)
        putchar(buffer[i]);
    putchar('\n');
}
```

```c
int main(void)
{
    char buffer[512];

    if( my_gets(buffer) != 0 ) {
        my_puts(buffer);
        printf("(%s)\n", buffer);
    }
    return 0;
}
```

입출력 결과

```
  abcd 123
  abcd 123
(  abcd 123)
계속하려면 아무 키나 누르십시오 . . .
```

입출력 결과

```
^Z
계속하려면 아무 키나 누르십시오 . . .
```

표준 문자 처리 함수

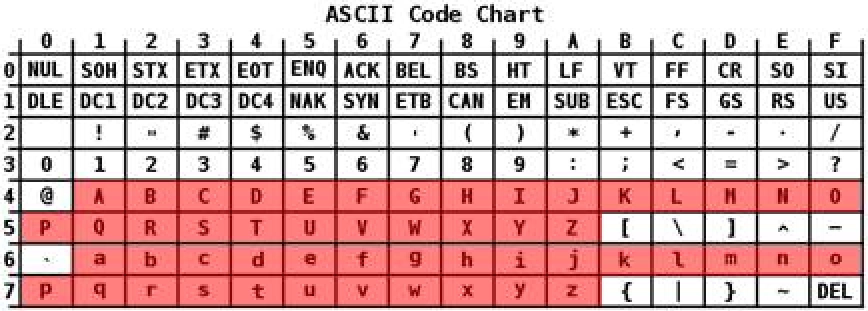
표준 문자 처리 함수 - <ctype.h>

함수	설명	함수	설명
int isalpha(int ch);	알파벳 확인	int isalnum(int ch);	알파벳,숫자 확인
int isupper(int ch);	대문자 확인	int isprint(int ch);	출력 가능 확인
int islower(int ch);	소문자 확인	int isgraph(int ch);	알파벳,숫자,구두점
int isdigit(int ch);	숫자 확인	int iscntrl(int ch);	제어 문자 확인
int isxdigit(int ch);	16진수 숫자 확인	int toupper(int ch);	대문자로 변환
int isspace(int ch);	공백 문자 확인	int tolower(int ch);	소문자로 변환
int ispunct(int ch);	구두 문자 확인		

표준 문자 처리 함수

- 알파벳 문자 확인

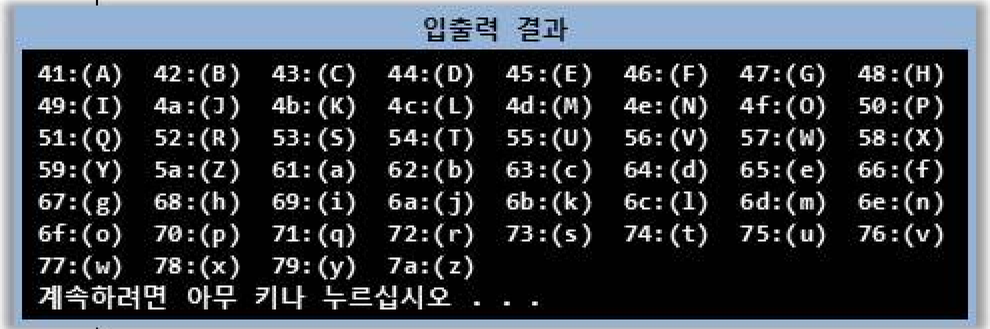
```
int isalpha(int ch);
```



ASCII Code Chart

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	NUL	SOH	STX	ETX	EOT	ENQ	ACK	BEL	BS	HT	LF	VT	FF	CR	SO	SI
1	DLE	DC1	DC2	DC3	DC4	NAK	SYN	ETB	CAN	EM	SUB	ESC	FS	GS	RS	US
2		!	"	#	$	%	&	'	(	)	*	+	,	-	.	/
3	0	1	2	3	4	5	6	7	8	9	:	;	<	=	>	?
4	@	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
5	P	Q	R	S	T	U	V	W	X	Y	Z	[	\	]	^	_
6	`	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o
7	p	q	r	s	t	u	v	w	x	y	z	{	|	}	~	DEL

```
#include <stdio.h>
#include <ctype.h>

int main(void)
{
    int ch;

    for(ch=0x00; ch<=0x7F; ++ch)
        if( isalpha(ch) )
            printf("%2x:(%c)\t",ch,ch);
    printf("\n");
    return 0;
}
```



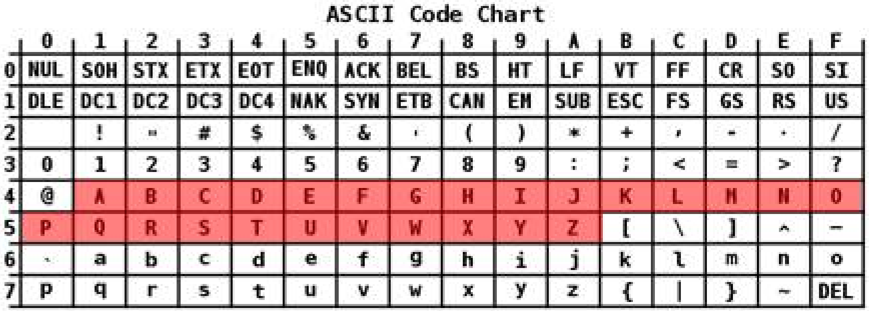
입출력 결과

```
41:(A)  42:(B)  43:(C)  44:(D)  45:(E)  46:(F)  47:(G)  48:(H)
49:(I)  4a:(J)  4b:(K)  4c:(L)  4d:(M)  4e:(N)  4f:(O)  50:(P)
51:(Q)  52:(R)  53:(S)  54:(T)  55:(U)  56:(V)  57:(W)  58:(X)
59:(Y)  5a:(Z)  61:(a)  62:(b)  63:(c)  64:(d)  65:(e)  66:(f)
67:(g)  68:(h)  69:(i)  6a:(j)  6b:(k)  6c:(l)  6d:(m)  6e:(n)
6f:(o)  70:(p)  71:(q)  72:(r)  73:(s)  74:(t)  75:(u)  76:(v)
77:(w)  78:(x)  79:(y)  7a:(z)
계속하려면 아무 키나 누르십시오 . . .
```

표준 문자 처리 함수

- 알파벳 대문자 확인

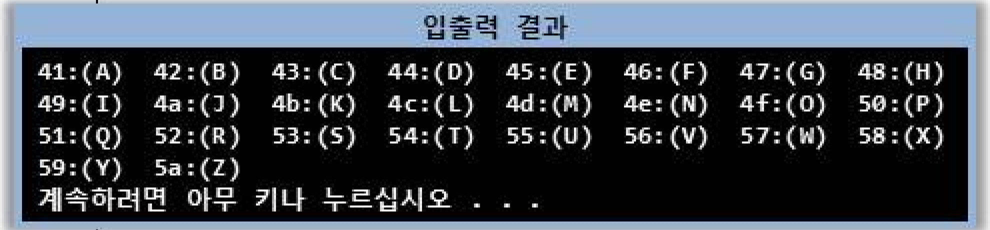
```
int isupper(int ch);
```



ASCII Code Chart

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	NUL	SOH	STX	ETX	EOT	ENQ	ACK	BEL	BS	HT	LF	VT	FF	CR	SO	SI
1	DLE	DC1	DC2	DC3	DC4	NAK	SYN	ETB	CAN	EM	SUB	ESC	FS	GS	RS	US
2		!	"	#	$	%	&	'	(	)	*	+	,	-	.	/
3	0	1	2	3	4	5	6	7	8	9	:	;	<	=	>	?
4	@	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
5	P	Q	R	S	T	U	V	W	X	Y	Z	[	\	]	^	_
6	`	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o
7	p	q	r	s	t	u	v	w	x	y	z	{	|	}	~	DEL

```
#include <stdio.h>
#include <ctype.h>

int main(void)
{
    int ch;

    for(ch=0x00; ch<=0x7F; ++ch)
        if( isupper(ch) )
            printf("%2x:(%c)\t",ch,ch);
    printf("\n");
    return 0;
}
```



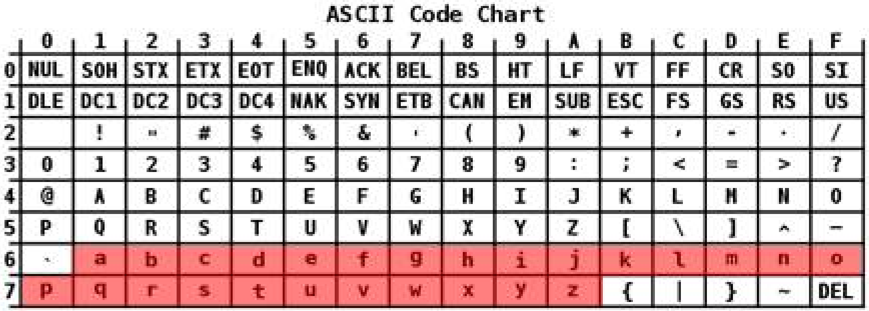
입출력 결과

```
41:(A)  42:(B)  43:(C)  44:(D)  45:(E)  46:(F)  47:(G)  48:(H)
49:(I)  4a:(J)  4b:(K)  4c:(L)  4d:(M)  4e:(N)  4f:(O)  50:(P)
51:(Q)  52:(R)  53:(S)  54:(T)  55:(U)  56:(V)  57:(W)  58:(X)
59:(Y)  5a:(Z)
계속하려면 아무 키나 누르십시오 . . .
```

표준 문자 처리 함수

- 알파벳 소문자 확인

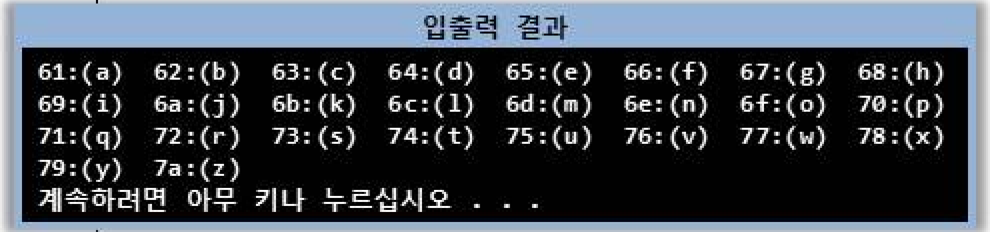
```
int islower(int ch);
```



ASCII Code Chart

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	NUL	SOH	STX	ETX	EOT	ENQ	ACK	BEL	BS	HT	LF	VT	FF	CR	SO	SI
1	DLE	DC1	DC2	DC3	DC4	NAK	SYN	ETB	CAN	EM	SUB	ESC	FS	GS	RS	US
2		!	"	#	$	%	&	'	(	)	*	+	,	-	.	/
3	0	1	2	3	4	5	6	7	8	9	:	;	<	=	>	?
4	@	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
5	P	Q	R	S	T	U	V	W	X	Y	Z	[	\	]	^	_
6	`	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o
7	p	q	r	s	t	u	v	w	x	y	z	{	|	}	~	DEL

```
#include <stdio.h>
#include <ctype.h>

int main(void)
{
    int ch;

    for(ch=0x00; ch<=0x7F; ++ch)
        if( islower(ch) )
            printf("%2x:(%c)\t",ch,ch);
    printf("\n");
    return 0;
}
```



입출력 결과

```
61:(a)  62:(b)  63:(c)  64:(d)  65:(e)  66:(f)  67:(g)  68:(h)
69:(i)  6a:(j)  6b:(k)  6c:(l)  6d:(m)  6e:(n)  6f:(o)  70:(p)
71:(q)  72:(r)  73:(s)  74:(t)  75:(u)  76:(v)  77:(w)  78:(x)
79:(y)  7a:(z)
계속하려면 아무 키나 누르십시오 . . .
```

표준 문자 처리 함수

- 숫자 확인

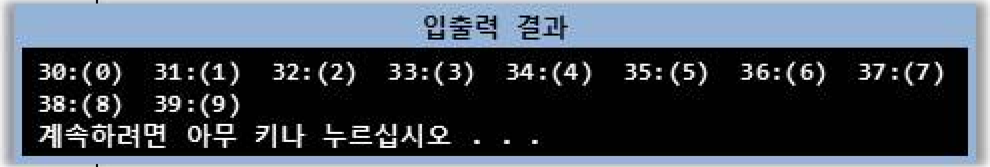
```
int isdigit(int ch);
```

ASCII Code Chart

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	NUL	SOH	STX	ETX	EOT	ENQ	ACK	BEL	BS	HT	LF	VT	FF	CR	SO	SI
1	DLE	DC1	DC2	DC3	DC4	NAK	SYN	ETB	CAN	EM	SUB	ESC	FS	GS	RS	US
2		!	"	#	$	%	&	'	(	)	*	+	,	-	.	/
3	0	1	2	3	4	5	6	7	8	9	:	;	<	=	>	?
4	@	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
5	P	Q	R	S	T	U	V	W	X	Y	Z	[	\	]	^	_
6	`	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o
7	p	q	r	s	t	u	v	w	x	y	z	{	|	}	~	DEL

```c
#include <stdio.h>
#include <ctype.h>

int main(void)
{
	int ch;

	for(ch=0x00; ch<=0x7F; ++ch)
		if( isdigit(ch) )
			printf("%2x:(%c)\t",ch,ch);
	printf("\n");
	return 0;
}
```

입출력 결과

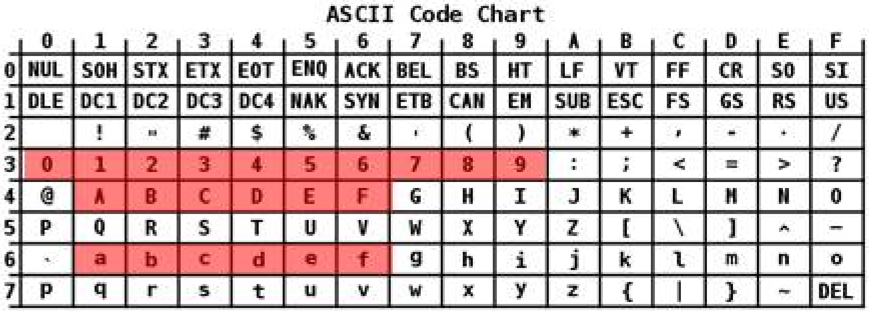


```
30:(0)  31:(1)  32:(2)  33:(3)  34:(4)  35:(5)  36:(6)  37:(7)
38:(8)  39:(9)
계속하려면 아무 키나 누르십시오 . . .
```

표준 문자 처리 함수

- 16진수 확인

```
int isxdigit(int ch);
```

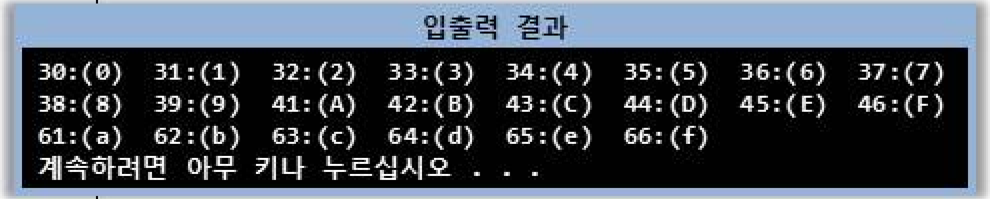


ASCII Code Chart

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	NUL	SOH	STX	ETX	EOT	ENQ	ACK	BEL	BS	HT	LF	VT	FF	CR	SO	SI
1	DLE	DC1	DC2	DC3	DC4	NAK	SYN	ETB	CAN	EM	SUB	ESC	FS	GS	RS	US
2		!	"	#	$	%	&	'	(	)	*	+	,	-	.	/
3	0	1	2	3	4	5	6	7	8	9	:	;	<	=	>	?
4	@	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
5	P	Q	R	S	T	U	V	W	X	Y	Z	[	\	]	^	_
6	`	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o
7	p	q	r	s	t	u	v	w	x	y	z	{	|	}	~	DEL

```
#include <stdio.h>
#include <ctype.h>

int main(void)
{
    int ch;

    for(ch=0x00; ch<=0x7F; ++ch)
        if( isxdigit(ch) )
            printf("%2x:(%c)\t",ch,ch);
    printf("\n");
    return 0;
}
```

입출력 결과

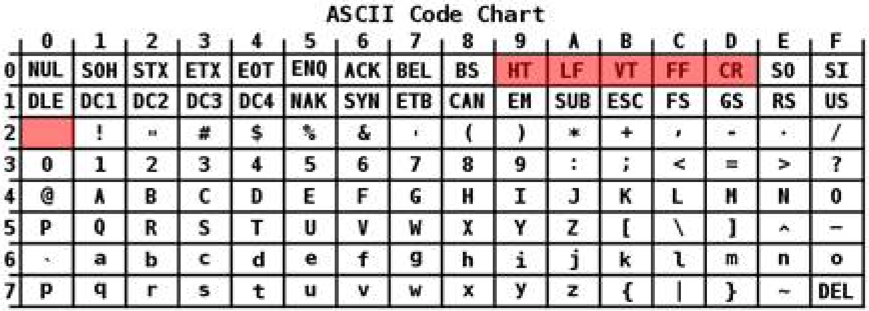


```
30:(0)   31:(1)   32:(2)   33:(3)   34:(4)   35:(5)   36:(6)   37:(7)
38:(8)   39:(9)   41:(A)   42:(B)   43:(C)   44:(D)   45:(E)   46:(F)
61:(a)   62:(b)   63:(c)   64:(d)   65:(e)   66:(f)
계속하려면 아무 키나 누르십시오 . . .
```

표준 문자 처리 함수

- 공백 문자 확인

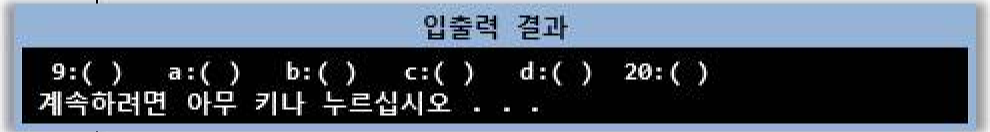
```
int isspace(int ch);
```



ASCII Code Chart

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	NUL	SOH	STX	ETX	EOT	ENQ	ACK	BEL	BS	HT	LF	VT	FF	CR	SO	SI
1	DLE	DC1	DC2	DC3	DC4	NAK	SYN	ETB	CAN	EM	SUB	ESC	FS	GS	RS	US
2		!	"	#	$	%	&	'	(	)	*	+	,	-	.	/
3	0	1	2	3	4	5	6	7	8	9	:	;	<	=	>	?
4	@	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
5	P	Q	R	S	T	U	V	W	X	Y	Z	[	\	]	^	_
6	`	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o
7	p	q	r	s	t	u	v	w	x	y	z	{	|	}	~	DEL

```
#include <stdio.h>
#include <ctype.h>

int main(void)
{
    int ch;

    for(ch=0x00; ch<=0x7F; ++ch)
        if( isspace(ch) )
            printf("%2x:(%c)\t",ch,' ');
    printf("\n");
    return 0;
}
```



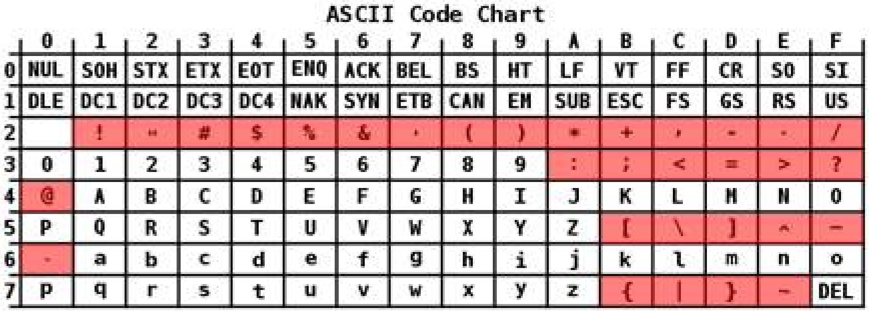
입출력 결과

```
 9:( )   a:( )   b:( )   c:( )   d:( )  20:( )
계속하려면 아무 키나 누르십시오 . . .
```

표준 문자 처리 함수

- 구두점 확인

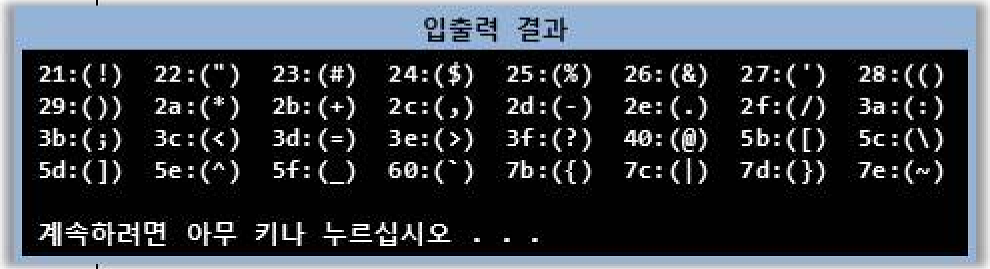
```
int ispunct(int ch);
```



ASCII Code Chart

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	NUL	SOH	STX	ETX	EOT	ENQ	ACK	BEL	BS	HT	LF	VT	FF	CR	SO	SI
1	DLE	DC1	DC2	DC3	DC4	NAK	SYN	ETB	CAN	EM	SUB	ESC	FS	GS	RS	US
2		!	"	#	$	%	&	'	(	)	*	+	,	-	.	/
3	0	1	2	3	4	5	6	7	8	9	:	;	<	=	>	?
4	@	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
5	P	Q	R	S	T	U	V	W	X	Y	Z	[	\	]	^	_
6	`	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o
7	p	q	r	s	t	u	v	w	x	y	z	{	|	}	~	DEL

```c
#include <stdio.h>
#include <ctype.h>

int main(void)
{
    int ch;

    for(ch=0x00; ch<=0x7F; ++ch)
        if( ispunct(ch) )
            printf("%2x:(%c)\t",ch,ch);
    printf("\n");
    return 0;
}
```



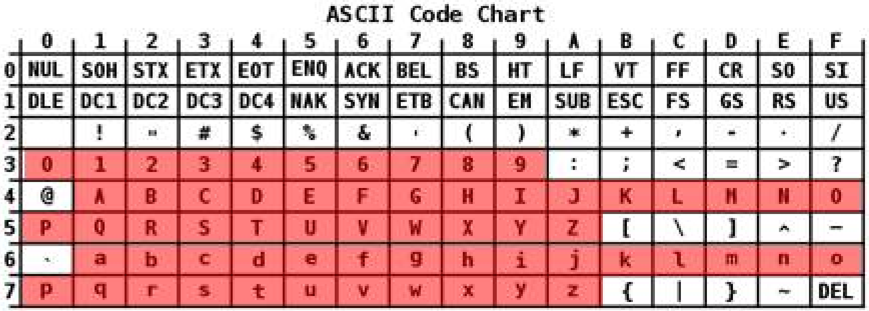
입출력 결과

```
21:(!)  22:(")  23:(#)  24:($)  25:(%)  26:(&)  27:(')  28:(()
29:())  2a:(*)  2b:(+)  2c:(,)  2d:(-)  2e:(.)  2f:(/)  3a:(:)
3b:(;)  3c:(<)  3d:(=)  3e:(>)  3f:(?)  40:(@)  5b:([)  5c:(\)
5d:(])  5e:(^)  5f:(_)  60:(`)  7b:({)  7c:(|)  7d:(})  7e:(~)

계속하려면 아무 키나 누르십시오 . . .
```

표준 문자 처리 함수

- 알파벳 및 숫자 확인

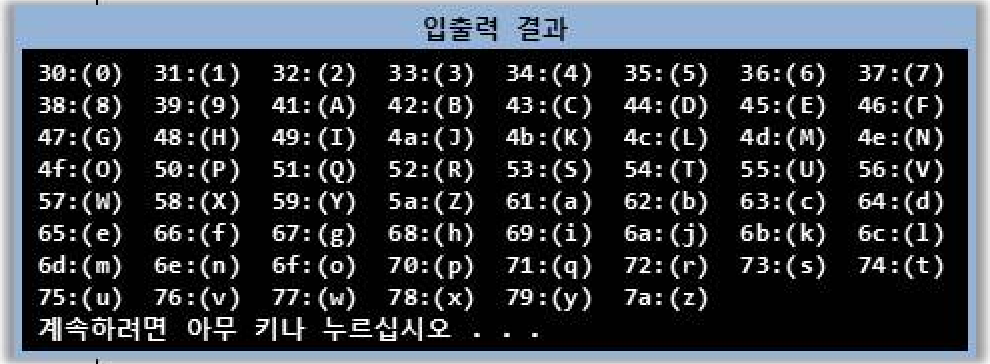
```
int isalnum(int ch);
```



ASCII Code Chart

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	NUL	SOH	STX	ETX	EOT	ENQ	ACK	BEL	BS	HT	LF	VT	FF	CR	SO	SI
1	DLE	DC1	DC2	DC3	DC4	NAK	SYN	ETB	CAN	EM	SUB	ESC	FS	GS	RS	US
2		!	"	#	$	%	&	'	(	)	*	+	,	-	.	/
3	0	1	2	3	4	5	6	7	8	9	:	;	<	=	>	?
4	@	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
5	P	Q	R	S	T	U	V	W	X	Y	Z	[	\	]	^	_
6	`	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o
7	p	q	r	s	t	u	v	w	x	y	z	{	|	}	~	DEL

```
#include <stdio.h>
#include <ctype.h>

int main(void)
{
    int ch;

    for(ch=0x00; ch<=0x7F; ++ch)
        if( isalnum(ch) )
            printf("%2x:(%c)\t",ch,ch);
    printf("\n");
    return 0;
}
```



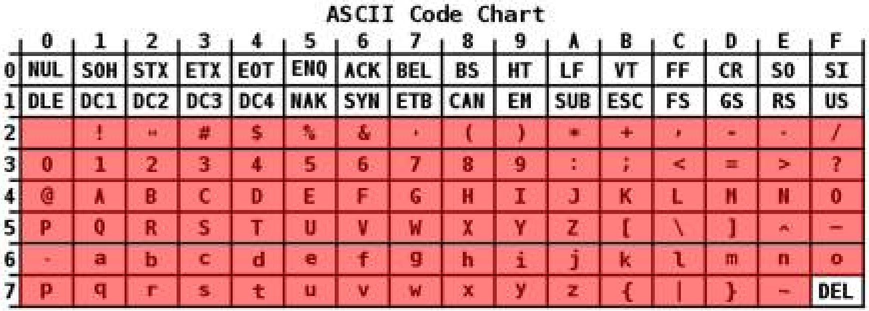
입출력 결과

```
30:(0)  31:(1)  32:(2)  33:(3)  34:(4)  35:(5)  36:(6)  37:(7)
38:(8)  39:(9)  41:(A)  42:(B)  43:(C)  44:(D)  45:(E)  46:(F)
47:(G)  48:(H)  49:(I)  4a:(J)  4b:(K)  4c:(L)  4d:(M)  4e:(N)
4f:(O)  50:(P)  51:(Q)  52:(R)  53:(S)  54:(T)  55:(U)  56:(V)
57:(W)  58:(X)  59:(Y)  5a:(Z)  61:(a)  62:(b)  63:(c)  64:(d)
65:(e)  66:(f)  67:(g)  68:(h)  69:(i)  6a:(j)  6b:(k)  6c:(l)
6d:(m)  6e:(n)  6f:(o)  70:(p)  71:(q)  72:(r)  73:(s)  74:(t)
75:(u)  76:(v)  77:(w)  78:(x)  79:(y)  7a:(z)
계속하려면 아무 키나 누르십시오 . . .
```

표준 문자 처리 함수

- 출력 가능 문자 확인

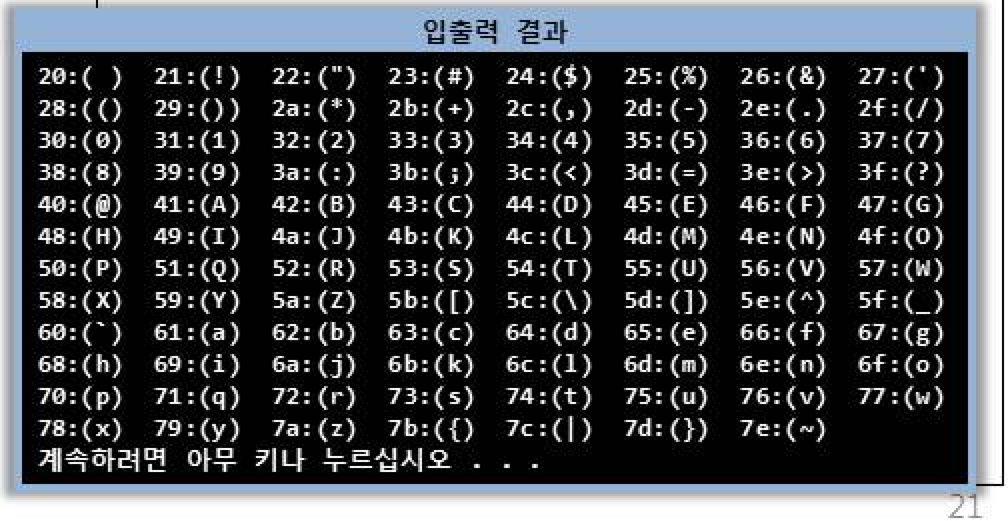
```
int isprint(int ch);
```



ASCII Code Chart

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	NUL	SOH	STX	ETX	EOT	ENQ	ACK	BEL	BS	HT	LF	VT	FF	CR	SO	SI
1	DLE	DC1	DC2	DC3	DC4	NAK	SYN	ETB	CAN	EM	SUB	ESC	FS	GS	RS	US
2		!	"	#	$	%	&	'	(	)	*	+	,	-	.	/
3	0	1	2	3	4	5	6	7	8	9	:	;	<	=	>	?
4	@	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
5	P	Q	R	S	T	U	V	W	X	Y	Z	[	\	]	^	_
6	`	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o
7	p	q	r	s	t	u	v	w	x	y	z	{	|	}	~	DEL

```
#include <stdio.h>
#include <ctype.h>

int main(void)
{
    int ch;

    for(ch=0x00; ch<=0x7F; ++ch)
        if( isprint(ch) )
            printf("%2x:(%c)\t",ch,ch);
    printf("\n");
    return 0;
}
```



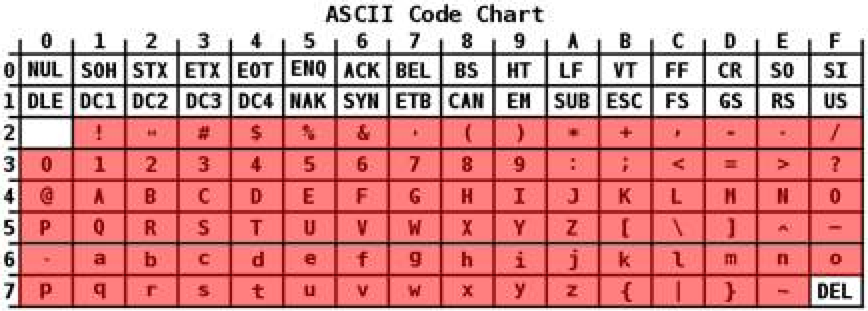
입출력 결과

```
20:( )   21:(!)   22:(")   23:(#)   24:($)   25:(%)   26:(&)   27:(')
28:(()   29:())   2a:(*)   2b:(+)   2c:(,)   2d:(-)   2e:(.)   2f:(/)
30:(0)   31:(1)   32:(2)   33:(3)   34:(4)   35:(5)   36:(6)   37:(7)
38:(8)   39:(9)   3a:(:)   3b:(;)   3c:(<)   3d:(=)   3e:(>)   3f:(?)
40:(@)   41:(A)   42:(B)   43:(C)   44:(D)   45:(E)   46:(F)   47:(G)
48:(H)   49:(I)   4a:(J)   4b:(K)   4c:(L)   4d:(M)   4e:(N)   4f:(O)
50:(P)   51:(Q)   52:(R)   53:(S)   54:(T)   55:(U)   56:(V)   57:(W)
58:(X)   59:(Y)   5a:(Z)   5b:([)   5c:(\)   5d:(])   5e:(^)   5f:(_)
60:(`)   61:(a)   62:(b)   63:(c)   64:(d)   65:(e)   66:(f)   67:(g)
68:(h)   69:(i)   6a:(j)   6b:(k)   6c:(l)   6d:(m)   6e:(n)   6f:(o)
70:(p)   71:(q)   72:(r)   73:(s)   74:(t)   75:(u)   76:(v)   77:(w)
78:(x)   79:(y)   7a:(z)   7b:({)   7c:(|)   7d:(})   7e:(~)
계속하려면 아무 키나 누르십시오 . . .
```

표준 문자 처리 함수

- 볼 수 있는 문자 확인

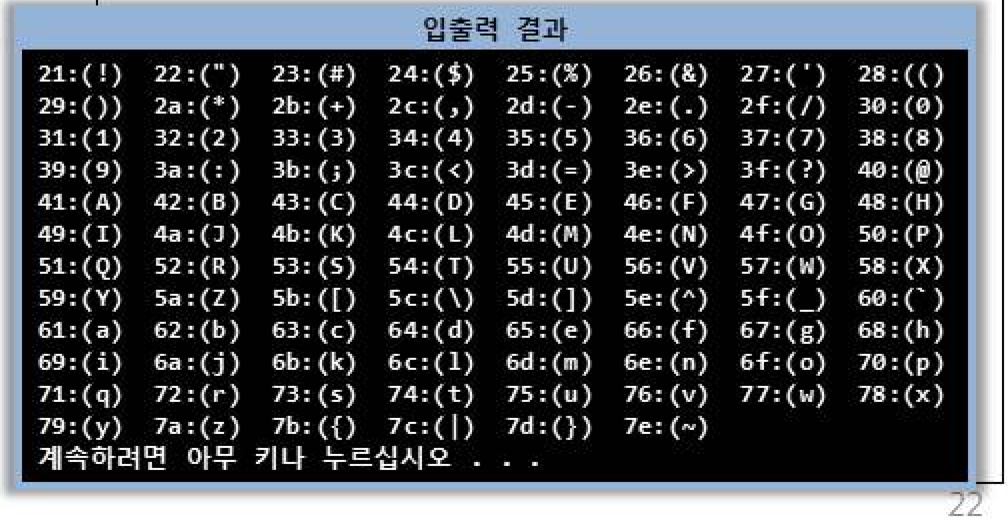
```
int isgraph(int ch);
```



ASCII Code Chart

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	NUL	SOH	STX	ETX	EOT	ENQ	ACK	BEL	BS	HT	LF	VT	FF	CR	SO	SI
1	DLE	DC1	DC2	DC3	DC4	NAK	SYN	ETB	CAN	EM	SUB	ESC	FS	GS	RS	US
2		!	"	#	$	%	&	'	(	)	*	+	,	-	.	/
3	0	1	2	3	4	5	6	7	8	9	:	;	<	=	>	?
4	@	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
5	P	Q	R	S	T	U	V	W	X	Y	Z	[	\	]	^	_
6	`	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o
7	p	q	r	s	t	u	v	w	x	y	z	{	|	}	~	DEL

```c
#include <stdio.h>
#include <ctype.h>

int main(void)
{
    int ch;

    for(ch=0x00; ch<=0x7F; ++ch)
        if( isgraph(ch) )
            printf("%2x:(%c)\t",ch,ch);
    printf("\n");
    return 0;
}
```



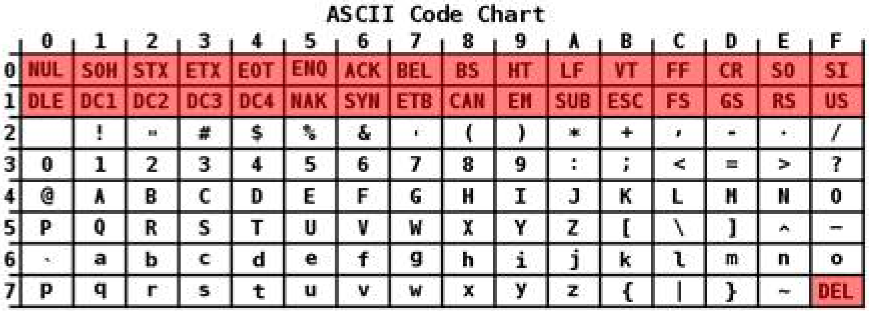
입출력 결과

```
21:(!)  22:(")  23:(#)  24:($)  25:(%)  26:(&)  27:(')  28:(()
29:())  2a:(*)  2b:(+)  2c:(,)  2d:(-)  2e:(.)  2f:(/)  30:(0)
31:(1)  32:(2)  33:(3)  34:(4)  35:(5)  36:(6)  37:(7)  38:(8)
39:(9)  3a:(:)  3b:(;)  3c:(<)  3d:(=)  3e:(>)  3f:(?)  40:(@)
41:(A)  42:(B)  43:(C)  44:(D)  45:(E)  46:(F)  47:(G)  48:(H)
49:(I)  4a:(J)  4b:(K)  4c:(L)  4d:(M)  4e:(N)  4f:(O)  50:(P)
51:(Q)  52:(R)  53:(S)  54:(T)  55:(U)  56:(V)  57:(W)  58:(X)
59:(Y)  5a:(Z)  5b:([)  5c:(\)  5d:(])  5e:(^)  5f:(_)  60:(`)
61:(a)  62:(b)  63:(c)  64:(d)  65:(e)  66:(f)  67:(g)  68:(h)
69:(i)  6a:(j)  6b:(k)  6c:(l)  6d:(m)  6e:(n)  6f:(o)  70:(p)
71:(q)  72:(r)  73:(s)  74:(t)  75:(u)  76:(v)  77:(w)  78:(x)
79:(y)  7a:(z)  7b:({)  7c:(|)  7d:(})  7e:(~)
계속하려면 아무 키나 누르십시오 . . .
```

표준 문자 처리 함수

- 제어 문자 확인

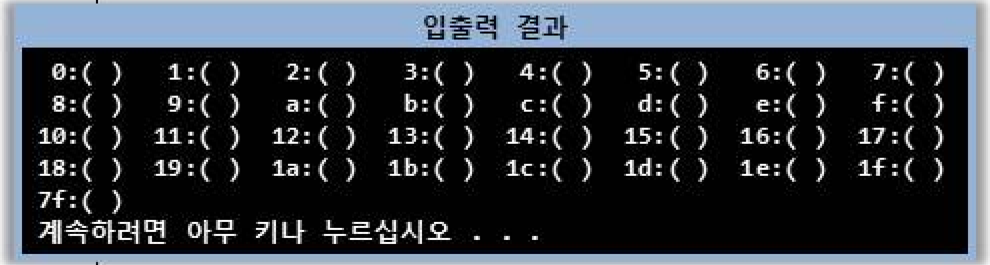
```
int iscntrl(int ch);
```



ASCII Code Chart

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	NUL	SOH	STX	ETX	EOT	ENQ	ACK	BEL	BS	HT	LF	VT	FF	CR	SO	SI
1	DLE	DC1	DC2	DC3	DC4	NAK	SYN	ETB	CAN	EM	SUB	ESC	FS	GS	RS	US
2		!	"	#	$	%	&	'	(	)	*	+	,	-	.	/
3	0	1	2	3	4	5	6	7	8	9	:	;	<	=	>	?
4	@	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
5	P	Q	R	S	T	U	V	W	X	Y	Z	[	\	]	^	_
6	`	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o
7	p	q	r	s	t	u	v	w	x	y	z	{	|	}	~	DEL

```
#include <stdio.h>
#include <ctype.h>

int main(void)
{
    int ch;

    for(ch=0x00; ch<=0x7F; ++ch)
        if( iscntrl(ch) )
            printf("%2x:(%c)\t",ch,' ');
    printf("\n");
    return 0;
}
```



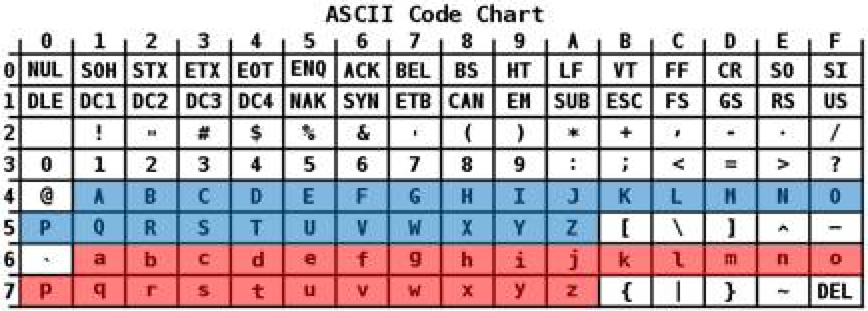
입출력 결과

```
 0:( )   1:( )   2:( )   3:( )   4:( )   5:( )   6:( )   7:( )
 8:( )   9:( )   a:( )   b:( )   c:( )   d:( )   e:( )   f:( )
10:( )  11:( )  12:( )  13:( )  14:( )  15:( )  16:( )  17:( )
18:( )  19:( )  1a:( )  1b:( )  1c:( )  1d:( )  1e:( )  1f:( )
7f:( )
계속하려면 아무 키나 누르십시오 . . .
```

표준 문자 처리 함수

- 소문자를 대문자로 전환

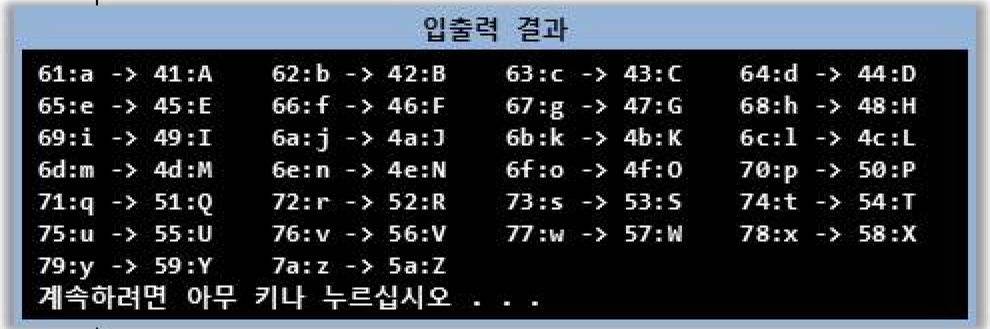
```
int toupper(int ch);
```



ASCII Code Chart

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	NUL	SOH	STX	ETX	EOT	ENQ	ACK	BEL	BS	HT	LF	VT	FF	CR	SO	SI
1	DLE	DC1	DC2	DC3	DC4	NAK	SYN	ETB	CAN	EM	SUB	ESC	FS	GS	RS	US
2		!	"	#	$	%	&	'	(	)	*	+	,	-	.	/
3	0	1	2	3	4	5	6	7	8	9	:	;	<	=	>	?
4	@	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
5	P	Q	R	S	T	U	V	W	X	Y	Z	[	\	]	^	_
6	`	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o
7	p	q	r	s	t	u	v	w	x	y	z	{	|	}	~	DEL

```c
#include <stdio.h>
#include <ctype.h>

int main(void)
{
    int ch;

    for(ch=0x61; ch<=0x7A; ++ch)
        printf("%2x:%c -> %2x:%c\t",
            ch, ch, toupper(ch), toupper(ch));
    printf("\n");
    return 0;
}
```



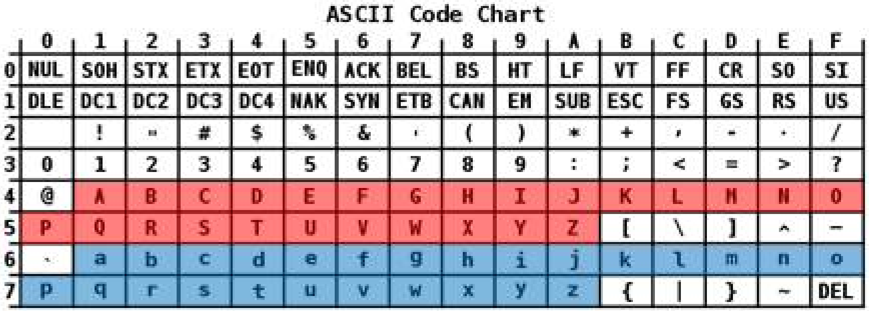
입출력 결과

```
61:a -> 41:A    62:b -> 42:B    63:c -> 43:C    64:d -> 44:D
65:e -> 45:E    66:f -> 46:F    67:g -> 47:G    68:h -> 48:H
69:i -> 49:I    6a:j -> 4a:J    6b:k -> 4b:K    6c:l -> 4c:L
6d:m -> 4d:M    6e:n -> 4e:N    6f:o -> 4f:O    70:p -> 50:P
71:q -> 51:Q    72:r -> 52:R    73:s -> 53:S    74:t -> 54:T
75:u -> 55:U    76:v -> 56:V    77:w -> 57:W    78:x -> 58:X
79:y -> 59:Y    7a:z -> 5a:Z
계속하려면 아무 키나 누르십시오 . . .
```

표준 문자 처리 함수

- 대문자를 소문자로 전환

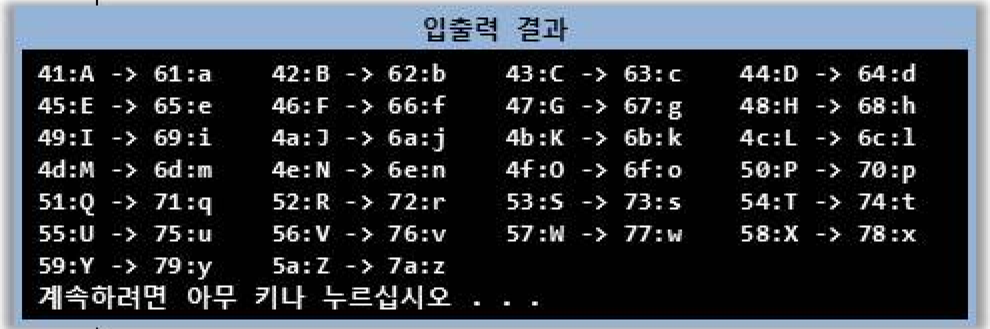
```
int tolower(int ch);
```



ASCII Code Chart

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	NUL	SOH	STX	ETX	EOT	ENQ	ACK	BEL	BS	HT	LF	VT	FF	CR	SO	SI
1	DLE	DC1	DC2	DC3	DC4	NAK	SYN	ETB	CAN	EM	SUB	ESC	FS	GS	RS	US
2		!	"	#	$	%	&	'	(	)	*	+	,	-	.	/
3	0	1	2	3	4	5	6	7	8	9	:	;	<	=	>	?
4	@	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
5	P	Q	R	S	T	U	V	W	X	Y	Z	[	\	]	^	_
6	`	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o
7	p	q	r	s	t	u	v	w	x	y	z	{	|	}	~	DEL

```c
#include <stdio.h>
#include <ctype.h>

int main(void)
{
    int ch;

    for(ch=0x41; ch<=0x5A; ++ch)
        printf("%2x:%c -> %2x:%c\t",
            ch, ch, tolower(ch), tolower(ch));
    printf("\n");
    return 0;
}
```



입출력 결과

```
41:A -> 61:a    42:B -> 62:b    43:C -> 63:c    44:D -> 64:d
45:E -> 65:e    46:F -> 66:f    47:G -> 67:g    48:H -> 68:h
49:I -> 69:i    4a:J -> 6a:j    4b:K -> 6b:k    4c:L -> 6c:l
4d:M -> 6d:m    4e:N -> 6e:n    4f:O -> 6f:o    50:P -> 70:p
51:Q -> 71:q    52:R -> 72:r    53:S -> 73:s    54:T -> 74:t
55:U -> 75:u    56:V -> 76:v    57:W -> 77:w    58:X -> 78:x
59:Y -> 79:y    5a:Z -> 7a:z
계속하려면 아무 키나 누르십시오 . . .
```

표준 문자열 처리 함수

표준 문자 처리 함수 - <string.h>

함수	설명
int strlen(char* str);	문자열의 길이를 반환한다.
char* strset(char* str,int ch);	문자열의 모든 문자들을 지정한 문자로 바꾼다.
char* strnset(char* str,int ch,int n);	문자열의 n개 문자를 지정한 문자로 바꾼다.
char* strcpy(char* dst,char* src);	문자열을 다른 문자열에 복사한다.
char* strcat(char* dst,char* src);	문자열을 다른 문자열에 덧붙인다.
char* strrev(char* str);	문자열을 뒤집는다.
int strcmp(char* str1,char* str2);	두 문자열을 비교한다.
char* strchr(char* str,int ch);	지정한 문자의 첫 번째 출현 위치를 찾는다.
int strcspn(char* str,char* set);	지정한 문자들의 첫 번째 출현 위치를 찾는다.
char* strstr(char* str,char* sub);	지정한 문자열의 첫 번째 출현 위치를 찾는다.

표준 문자열 처리 함수의 사용

- strlen( ) 함수
  - Use
    - Find length of string
  - Prototype

```
int strlen(char* str);
```

  - Return
    - Each of these functions returns the number of characters in *str*, excluding the terminal NULL(0).

입출력 결과

```
4
8
계속하려면 아무 키나 누르십시오 . . .
```

```
#include <stdio.h>
#include <string.h>

int main(void)
{
    char a[5]="abcd";
    char b[]="1234abcd";

    printf("%d\n", strlen(a) );
    printf("%d\n", strlen(b) );
    return 0;
}
```

표준 문자열 처리 함수의 사용

- strset( ) 함수
  - Use
    - Set all characters of string to specified character
  - Prototype

```
char* strset(char* str,int ch);
```

  - Return
    - Returns a pointer to the altered string *str.*

입출력 결과

```
eeee
11111111
계속하려면 아무 키나 누르십시오 . . .
```

```
#include <stdio.h>
#include <string.h>

int main(void)
{
    char a[5]="abcd";
    char b[]="1234abcd";

    printf("%s\n", strset(a,'e') );
    printf("%s\n", strset(b,'1') );
    return 0;
}
```

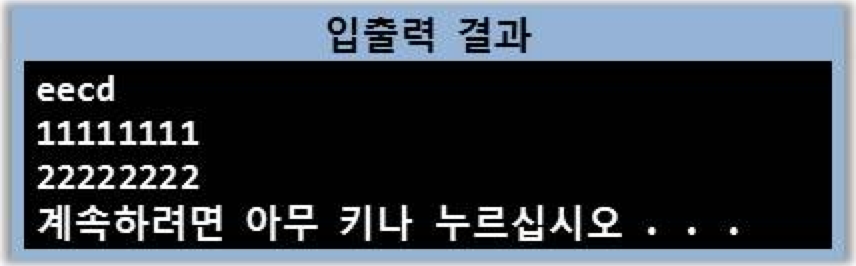
표준 문자열 처리 함수의 사용

- strnset( ) 함수
  - Use
    - Set first n characters of string to specified character
  - Prototype

```
char* strnset(char* str,int ch,int n);
```

  - Return
    - Returns a pointer to the altered string *str*.

입출력 결과



```
eecd
11111111
22222222
계속하려면 아무 키나 누르십시오 . . .
```

```
#include <stdio.h>
#include <string.h>

int main(void)
{
	char a[5]="abcd";
	char b[]="1234abcd";

	printf("%s\n", strnset(a,'e',2) );
	printf("%s\n", strnset(b,'1',8) );
	printf("%s\n", strnset(b,'2',10) );
	return 0;
}
```

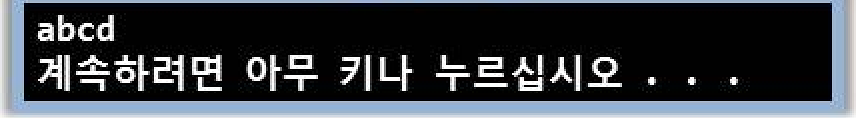
표준 문자열 처리 함수의 사용

- strcpy( ) 함수
  - Use
    - Copy one string to another
  - Prototype

```
char* strcpy(char* dst,char* src);
```

  - Return
    - Each of these functions returns the destination string *dst*.

입출력 결과



```
abcd
계속하려면 아무 키나 누르십시오 . . .
```

```
#include <stdio.h>
#include <string.h>

int main(void)
{
    char a[5]="abcd";
    char b[]="1234abcd";

    printf("%s\n", strcpy(b,a) );
    return 0;
}
```

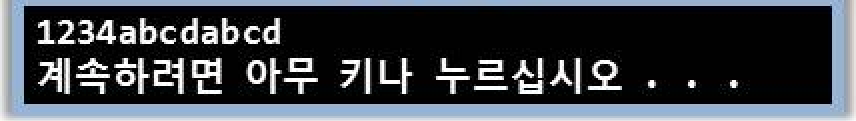
표준 문자열 처리 함수의 사용

- strcat( ) 함수
  - Use
    - Append one string to another
  - Prototype

```
char* strcat(char* dst,char* src);
```

  - Return
    - Each of these functions returns the destination string *dst*.

입출력 결과



```
1234abcdabcd
계속하려면 아무 키나 누르십시오 . . .
```

```
#include <stdio.h>
#include <string.h>

int main(void)
{
    char a[5]="abcd";
    char b[13]="1234abcd";

    printf("%s\n", strcat(b,a) );
    return 0;
}
```

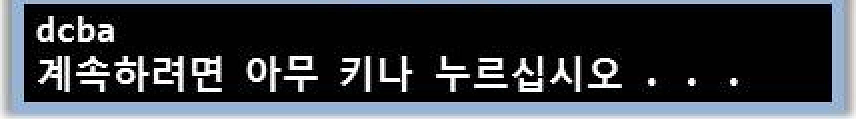
표준 문자열 처리 함수의 사용

- strrev( ) 함수
  - Use
    - Reverse string
  - Prototype

```
char* strrev(char* str);
```

  - Return
    - Returns a pointer to the altered string.

입출력 결과



```
dcba
계속하려면 아무 키나 누르십시오 . . .
```

```
#include <stdio.h>
#include <string.h>

int main(void)
{
    char a[5]="abcd";

    printf("%s\n", strrev(a) );
    return 0;
}
```

표준 문자열 처리 함수의 사용

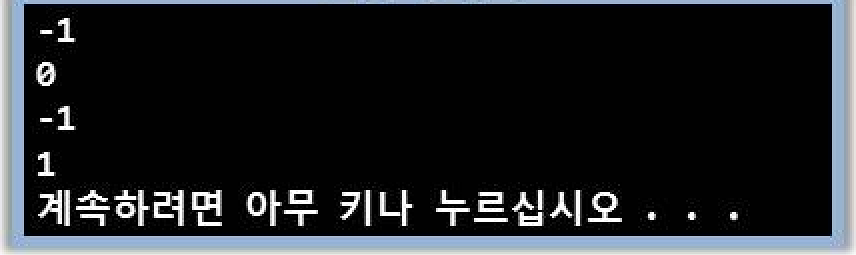
- strcmp( ) 함수
  - Use
    - Compare two strings
  - Prototype

```
int strcmp(char* str1,char* str2);
```

  - Return
    - ret < 0: *str1* less than *str2*
    - ret = 0: *str1* identical to *str2*
    - ret > 0: *str1* greater than *str2*

```c
#include <stdio.h>
#include <string.h>

int main(void)
{
    char a[10]="abc";
    char b[10]="abcd";
    char c[10]="abcde";
    char d[10]="aaaa";

    printf("%d\n", strcmp(a,b) );
    printf("%d\n", strcmp(b,b) );
    printf("%d\n", strcmp(b,c) );
    printf("%d\n", strcmp(b,d) );
    return 0;
}
```

입출력 결과



```
-1
0
-1
1
계속하려면 아무 키나 누르십시오 . . .
```

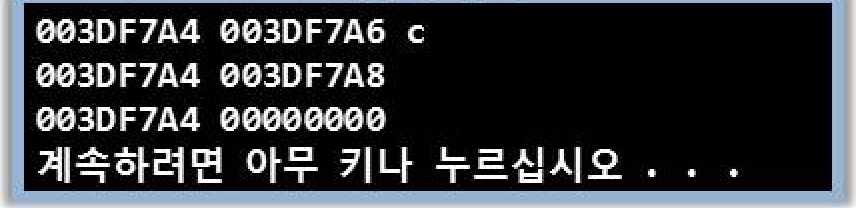
표준 문자열 처리 함수의 사용

- strchr( ) 함수
  - Use
    - Find first occurrence of specified character in string
  - Prototype

```
char* strchr(char* str,int ch);
```

  - Return
    - Each of these functions returns a pointer to the first occurrence of *ch* in *str*, or NULL(0) if *ch* is not found.

입출력 결과



```
003DF7A4 003DF7A6 c
003DF7A4 003DF7A8
003DF7A4 00000000
계속하려면 아무 키나 누르십시오 . . .
```

```
#include <stdio.h>
#include <string.h>

int main(void)
{
    char a[10]="abcd";
    char* p;

    p=strchr(a,'c');
    printf("%p %p %c\n", a, p, *p);
    p=strchr(a,'\0');
    printf("%p %p %c\n", a, p, ' ');
    p=strchr(a,'X');
    printf("%p %p %c\n", a, p, ' ');
    return 0;
}
```

표준 문자열 처리 함수의 사용

- strcspn( ) 함수
  - Use
    - Find first occurrence of character from specified character set in string
  - Prototype

```
int strcspn(char* str,char* set);
```

  - Return
    - These functions return the index of the first character in *str* that is in *set*. If none of the characters in *str* is in *set*, then the return value is the length of *str*.

입출력 결과

```
3 d
계속하려면 아무 키나 누르십시오 . . .
```

```
#include <stdio.h>
#include <string.h>

int main(void)
{
    char a[10]="abcdefghi";
    int i;

    i = strcspn(a,"fd");
    printf("%d %c\n", i, a[i] );
    return 0;
}
```

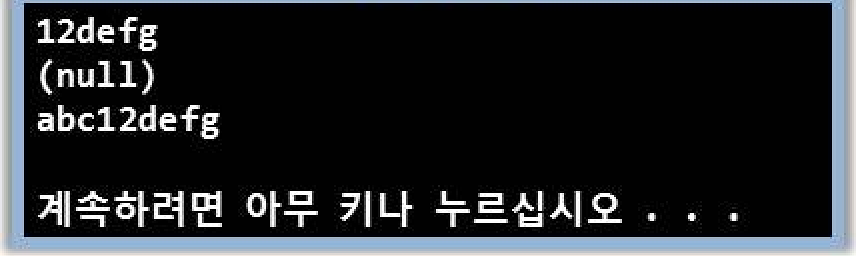
표준 문자열 처리 함수의 사용

- strstr( ) 함수
  - Use
    - Find first occurrence of specified string in another string
  - Prototype

```
char* strstr(char* str,char* sub);
```

  - Return
    - Returns a pointer to the first occurrence of *sub* in *str*, or NULL(0) if *sub* does not appear in *str*.

입출력 결과



```
12defg
(null)
abc12defg

계속하려면 아무 키나 누르십시오 . . .
```

```
#include <stdio.h>
#include <string.h>

int main(void)
{
    char a[20]="abc12defg";

    printf("%s\n", strstr(a,"12") );
    printf("%s\n", strstr(a,"XY") );
    printf("%s\n", strstr(a,"") );
    printf("%s\n", strstr("","") );
    return 0;
}
```