

2번

```
#include <iostream>
using namespace std;

int main() {
    char ch;
    int count = 0;
    cout << "한 라인 입력: ";
    while (true) {
        cin.get(ch); // 키를 ch에 읽어옴
        if (cin.eof()) break; // EOF 문자 즉 ctrl-z키가 입력된 경우, 입력 종료
        cout.put(ch); // 읽은 문자 출력
        if (ch == '\n') { // <Enter> 키가 입력된 경우 읽기 중단
            break;
        }
        else if (ch == ' ') // ' ' (빈칸)이 입력된 경우
            count++; // count 증가
    }
    cout << count << endl; // count값 출력
    return 0;
}
```

C:\WINDOWS\system32\cmd.exe

```
한 라인 입력: C++ is too hard.
C++ is too hard.
3
Press any key to continue . . .
```

-----다음 페이지에 7,9번 풀이

## 7번

```
#include <iostream>
#include <cctype> // 문제에서 주어진 헤더파일
#include <iomanip> // 매개변수를 가지는 조작자를 쓰기 위해 <iomanip> 헤더파일을 include
using namespace std;

int main() {
    cout.setf(ios::left); // 오른쪽 정렬이 디폴트이므로 왼쪽 맞춤으로 출력설정
    for (int i = 0; i < 4; i++) { // 총 4번 출력되게 해야 함
        cout << setw(7) << "dec"; // 7칸 너비에 "dec" 출력
        cout << setw(7) << "hexa"; // 7칸 너비에 "hexa" 출력
        cout << setw(7) << "char"; // 7칸 너비에 "char" 출력
    }
    cout << endl;

    for (int i = 0; i < 4; i++) { // 총 4번 출력되게 해야 함
        cout << setw(7) << "____"; // 7칸 너비에 ____ 출력
        cout << setw(7) << "____"; // 7칸 너비에 ____ 출력
        cout << setw(7) << "____"; // 7칸 너비에 ____ 출력
    }
    cout << endl;

    for (int i = 0; i < 128; i++) { // 0 ~ 127번까지 ASCII 코드와 해당 문자를 출력
        if (i != 0) { // i가 0인 경우 줄바꿈을 바꾸면 안됨.
            if ((i % 4) == 0) { // i가 4의 배수 단위마다 줄바꿈 (위에 for문에서 총 4set으로 구성했기 때문)
                cout << endl;
            }

            cout << setw(7) << dec << i; // i의 값을 10진수로 출력
            cout << setw(7) << hex << i; // i의 값을 16진수로 출력
            //char 같은 경우 출력불가능 ASCII코드 고려해야 한다.
            //문자가 출력 가능한지 알기 위해 bool isprint(int c); 함수 사용
            //isprint함수는 출력 문자의 아스키코드 판별 (출력 가능이면 0이 아닌수, 출력 불가능이면 0 리턴)
            if (isprint(i) == 0) {
                cout << setw(7) << '.'; // 출력 불가능이므로 . 출력
            }
            else { // 출력 가능이면
                cout << setw(7) << (char)i; // 정수형 char로 형변환
            }
        }

        cout << endl;
    }
    return 0;
}
```

C:\WINDOWS\system32\cmd.exe

dec	hexa	char	dec	hexa	char	dec	hexa	char	dec	hexa	char
0	0	.	1	1	.	2	2	.	3	3	.
4	4	.	5	5	.	6	6	.	7	7	.
8	8	.	9	9	.	10	a	.	11	b	.
12	c	.	13	d	.	14	e	.	15	f	.
16	10	.	17	11	.	18	12	.	19	13	.
20	14	.	21	15	.	22	16	.	23	17	.
24	18	.	25	19	.	26	1a	.	27	1b	.
28	1c	.	29	1d	.	30	1e	.	31	1f	.
32	20	.	33	21	!	34	22	"	35	23	#
36	24	\$	37	25	%	38	26	&	39	27	'
40	28	(	41	29	)	42	2a	*	43	2b	+
44	2c	,	45	2d	-	46	2e	.	47	2f	/
48	30	0	49	31	1	50	32	2	51	33	3
52	34	4	53	35	5	54	36	6	55	37	7
56	38	8	57	39	9	58	3a	:	59	3b	;
60	3c	<	61	3d	=	62	3e	>	63	3f	?
64	40	@	65	41	A	66	42	B	67	43	C
68	44	D	69	45	E	70	46	F	71	47	G
72	48	H	73	49	I	74	4a	J	75	4b	K
76	4c	L	77	4d	M	78	4e	N	79	4f	O
80	50	P	81	51	Q	82	52	R	83	53	S
84	54	T	85	55	U	86	56	V	87	57	W
88	58	X	89	59	Y	90	5a	Z	91	5b	[
92	5c	\	93	5d	]	94	5e	^	95	5f	_
96	60	`	97	61	a	98	62	b	99	63	`
100	64	d	101	65	e	102	66	f	103	67	g
104	68	h	105	69	i	106	6a	j	107	6b	k
108	6c	l	109	6d	m	110	6e	n	111	6f	o
112	70	p	113	71	q	114	72	r	115	73	s
116	74	t	117	75	u	118	76	v	119	77	w
120	78	x	121	79	y	122	7a	z	123	7b	{
124	7c		125	7d	}	126	7e	~	127	7f	.

Press any key to continue . . .



## 9번

```
#include <iostream>
using namespace std;

class Phone { // 전화번호를 표현하는 클래스
    string name;
    string telnum;
    string address;
public:
    Phone(string name = "", string telnum = "", string address = "") {
        this->name = name;
        this->telnum = telnum;
        this->address = address;
    }
    // 사용자 삽입 연산자 // << 연산자가 참조 리턴
    friend ostream& operator << (ostream& stream, Phone p); // friend 선언(private필드에 접근하기 위함)
    // 사용자 추출 연산자 // 참조 리턴
    friend istream& operator >> (istream& ins, Phone& ph); // friend 선언(private필드에 접근하기 위함)
};

//이들 연산자 함수는 외부 함수로 작성되어야 한다.
// 사용자 삽입 연산자 구현
ostream& operator << (ostream& stream, Phone p) {
    stream << "(" << p.name << "," << p.telnum << "," << p.address << ")";
    return stream; // 참조리턴
}

// 사용자 추출 연산자 구현
istream& operator >> (istream& ins, Phone& ph) { // 두 번째 매개변수를 참조 타입으로 선언하는 이유:
    // 원본 객체에 값을 쓰기 위함.
    cout << "이름:";
    ins >> ph.name;
    cout << "전화번호:";
    ins >> ph.telnum;
    cout << "주소:";
    ins >> ph.address;
    return ins; //참조리턴
}

int main() {
    Phone girl, boy;
    cin >> girl >> boy; // 전화 번호를 키보드로부터 읽는다.
    cout << girl << endl << boy << endl; // 전화 번호를 출력한다.
    return 0;
}
```

C:\WINDOWS\system32\cmd.exe

```
이름:Girls'Generation
전화번호:010-0000-0000
주소:Girls'Home
이름:PSY
전화번호:010-1111-1111
주소:PSY'Home
(Girls'Generation,010-0000-0000,Girls'Home)
(PSY,010-1111-1111,PSY'Home)
Press any key to continue
```