논리 연산자

▣ 논리 연산자

구분	연산식			결과	설명
AND (논리곱)	true	&& 또는 &	true	true	피 연사자 모두가 true 일 경우에만 연산 결과는 true
	true		false	false	
	false		true	false	
	false		false	false	
OR (논리합)	true	|| 또는 |	true	true	피 연산자 중 하나만 true 이면 연산 결과는 true
	true		false	true	
	false		true	ture	
	false		false	false	
XOR (배타적 논리합)	true	^	true	false	피 연산자가 하나는 ture 이고 다른 하나가 false 일 경우에만 연산 결과는 true
	true		false	true	
	false		true	ture	
	false		false	false	
NOT (논리부정)		!	true	false	피 연산자의 논리값을 바꿈
			false	true	

&&와 &는 산출 결과는 같지만 연산 과정이 조금 다르다.
&&는 앞의 피연산자가 false라면 뒤의 피연산자를 평가하지 않고 바로 false를 산출한다.
그러나 &는 두 피연산자 모두를 평가해서 산출 결과를 낸다. 따라서 &보다 &&가 더 효율적으로 동작한다.
||과 |도 마찬가지입니다.

① x는 10보다 크고, 20보다 작다.

'x 〉 10'와 'x 〈 20'가 '그리고(and)'로 연결된 조건이므로 다음과 같이 쓸 수 있다.

```
x > 10 && x < 20
```

'x 〉 10'는 '10 〈 x'와 같으므로 다음과 같이 쓸 수도 있다. 보통은 변수를 왼쪽에 쓰지만 이런 경우 가독성측면에서 보면 아래의 식이 더 나을 수 있다. (권장)

```
10 < x && x < 20
```

※ 10 < x < 20 //잘못된 식

② i는 2의 배수 또는 3의 배수이다.

어떤 수가 2의 배수라는 얘기는 2로 나누었을 때 나머지가 0이라는 뜻이다. 그래서 나머지 연산의 결과가 0인지 확인하면 된다. '또는'으로 두 조건이 연결되었으므로 논리 연산자 '||'(OR)를 사용해야 한다.

```
i%2==0 || i%3==0
```

i의 값이 8일 때, 위의 식은 다음과 같은 과정으로 연산된다.

```
   i%2==0 || i%3==0
→  8%2==0 || 8%3==0
→    0==0 || 2==0
→    true || false
→         true
```

③ i는 2의 배수 또는 3의 배수지만 6의 배수는 아니다.

이전 조건에 6의 배수를 제외하는 조건이 더 붙었다. 6의 배수가 아니어야 한다는 조건은 'i%6!=0'이고, 이 조건을 '&&(AND)'로 연결해야 한다.

```
( i%2==0 || i%3==0 ) && i%6!=0
```

위의 식에 괄호를 사용한 이유는 '&&'가 '||'보다 우선순위가 높기 때문이다. 만일 괄호를 사용하지 않으면 '&&'를 먼저 연산한다. 다음의 두 식은 동일하다.

```
i%2==0 ||  i%3==0 && i%6!=0
i%2==0 || (i%3==0 && i%6!=0)
```

④ 문자 ch는 숫자('0'~'9')이다.

사용자로부터 입력된 문자가 숫자('0'~'9')인지 확인하는 식은 다음과 같이 쓸 수 있다.

```
'0' <= ch && ch <= '9'
```

유니코드에서 문자 '0'부터 '9'까지 연속적으로 배치되어 있기 때문에 가능한 식이다. 문자 '0'부터 '9'까지 유니코드는 10진수로 다음과 같다.

문자	'0'	'1'	'2'	'3'	'4'	'5'	'6'	'7'	'8'	'9'
문자코드	48	49	50	51	52	53	54	55	56	57

⑤ 문자 ch는 대문자 또는 소문자이다.

④의 경우와 마찬가지로 문자 'a'부터 'z'까지, 그리고 'A'부터 'Z'까지도 연속적으로 배치되어 있으므로 문자 ch가 대문자 또는 소문자'인지 확인하는 식은 다음과 같이 쓸 수 있다.

```
('a' <= ch && ch <= 'z') || ('A' <= ch && ch <= 'Z')
```

아스키코드 값은
대문자A는 65이며 대문자 Z는 90이며,
소문자 a는 97이고 소문자 z는 122이다.

대소문자 아스키코드 값 차이는 32이다.

대문자에서 소문자로 변환 할때는 32를 더해주고,
소문자에서 대문자로 변환할때는 32를 빼주면 된다.