

# 프로그램 논리(2)

---



인하공업전문대학 컴퓨터정보과

이수정 교수

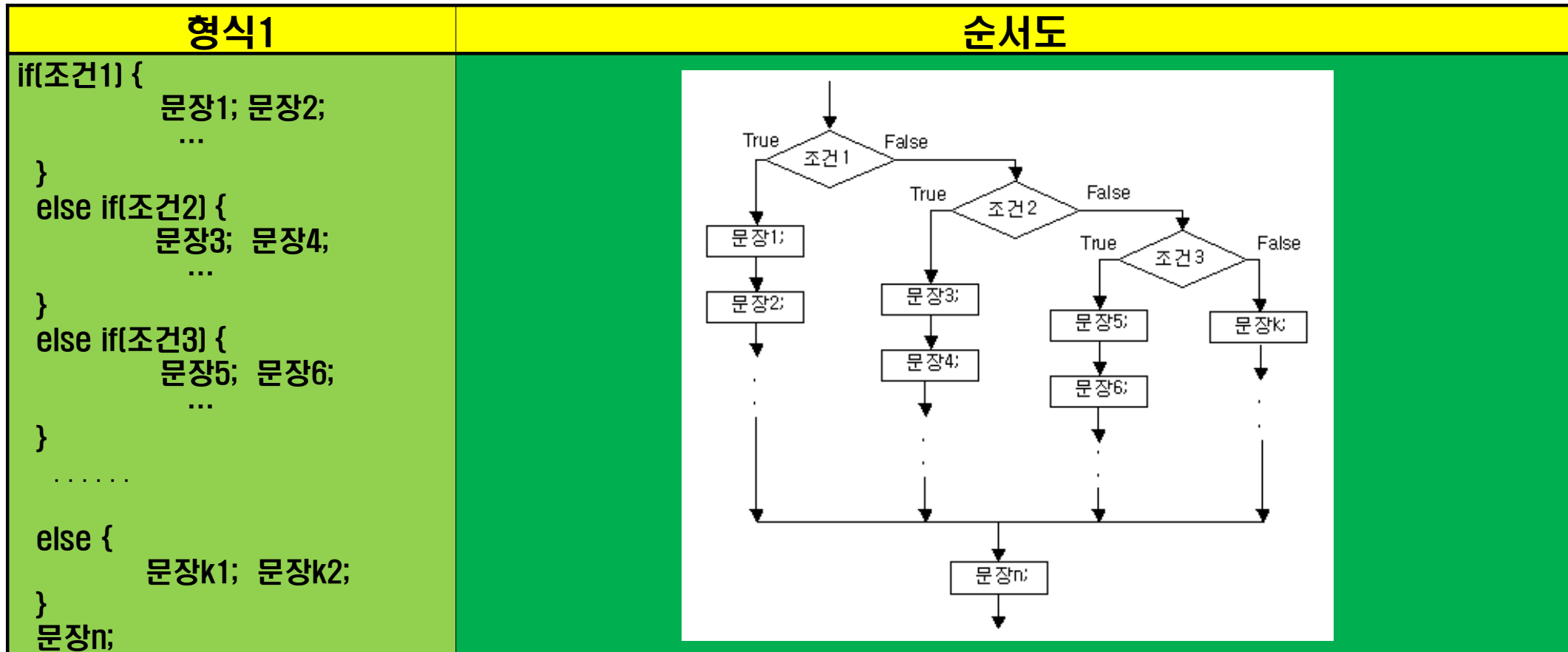


## 차례

- 프로그램 논리
  1. 순차 논리
  2. 선택 논리
  3. 반복 논리



## 2.3 다중선택 if문



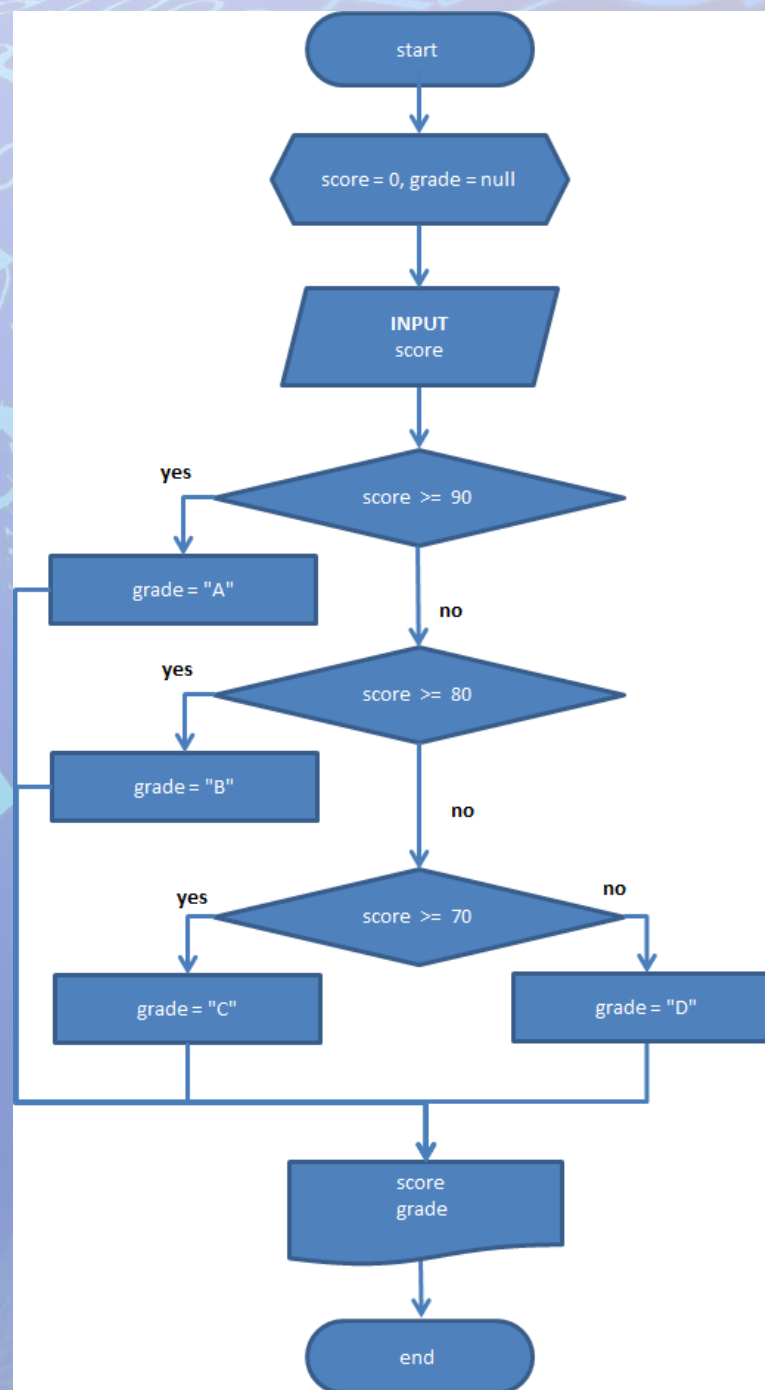
조건1을 판별해서 조건1이 참(true)이면 if 블록 안의 문장들(문장1, 문장2, ...)을 수행하고 나서 문장n을 수행한다. 그렇지 않고 조건1이 거짓(false)이면 다시 조건2를 판별한다. 판별해서 조건2가 참이면 else if 블록 안의 문장들(문장3, 문장4, ...)을 수행하고 나서 문장n을 수행한다. 그렇지 않고 조건2도 거짓(false)이면 다시 조건3을 판별한다. 판별해서 조건3이 참이면 else if 블록 안의 문장들(문장5, 문장6, ...)을 수행하고 나서 문장n을 수행한다. 그렇지 않고 조건3도 거짓이면 else 블록 내의 문장들(문장k1, 문장k2, ...)을 수행하고 나서 문장n을 수행하게 된다. 이 형식에서 else if는 필요시 얼마든지 더 나타낼 수 있다.

# 실습 예제

- 0부터 100 사이의 임의의 숫자(과목 점수라고 가정)를 하나 읽어 들여 이 점수의 학점이 무엇인지 출력하는 순서도를 작성하여라. 과목 점수와 학점의 관계는 다음과 같은 조건으로 한다.
- ( 조건 )
  - A학점 : 점수  $\geq 90$
  - B학점 :  $90 > \text{점수} \geq 80$
  - C학점 :  $80 > \text{점수} \geq 70$
  - D학점 :  $70 > \text{점수}$

## [문제해결논리]

학점별 조건이 다중선택 if문의 조건으로 표현되어야 하며, 각 조건이 만족되었을 때 해당되는 학점을 부여하면 된다. 또한 if문이 끝나면 입력된 점수와 부여된 학점을 인쇄하면 된다.



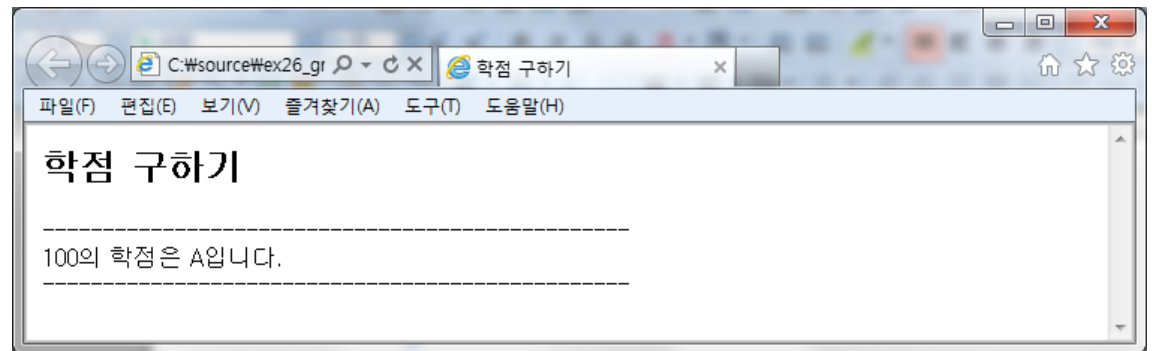
```
ex26_grade - 메모장
파일(F) 편집(E) 서식(O) 보기(V) 도움말(H)

<html>
<head> <title> 학점 구하기 </title> </head>
<body>
  <script type = "text/javascript">
    document.write("<h2>학점 구하기</h2>");
    document.write("-----<br>");

    //점수에 대해 학점을 구하는 프로그램
    //100-90(A), 89-80(B), 79-70(C), 69-0(D)
    var score = 0; grade = null;
    score = prompt("점수를 입력하세요", "");

    if(score >= 90)    { grade = "A"; }
    else if(score >= 80) { grade = "B"; }
    else if(score >= 70) { grade = "C"; }
    else              { grade = "D"; }

    document.write(score + "의 학점은 " + grade + "입니다.<br>");
    document.write("-----<br>");
  </script>
</body>
</html>
```



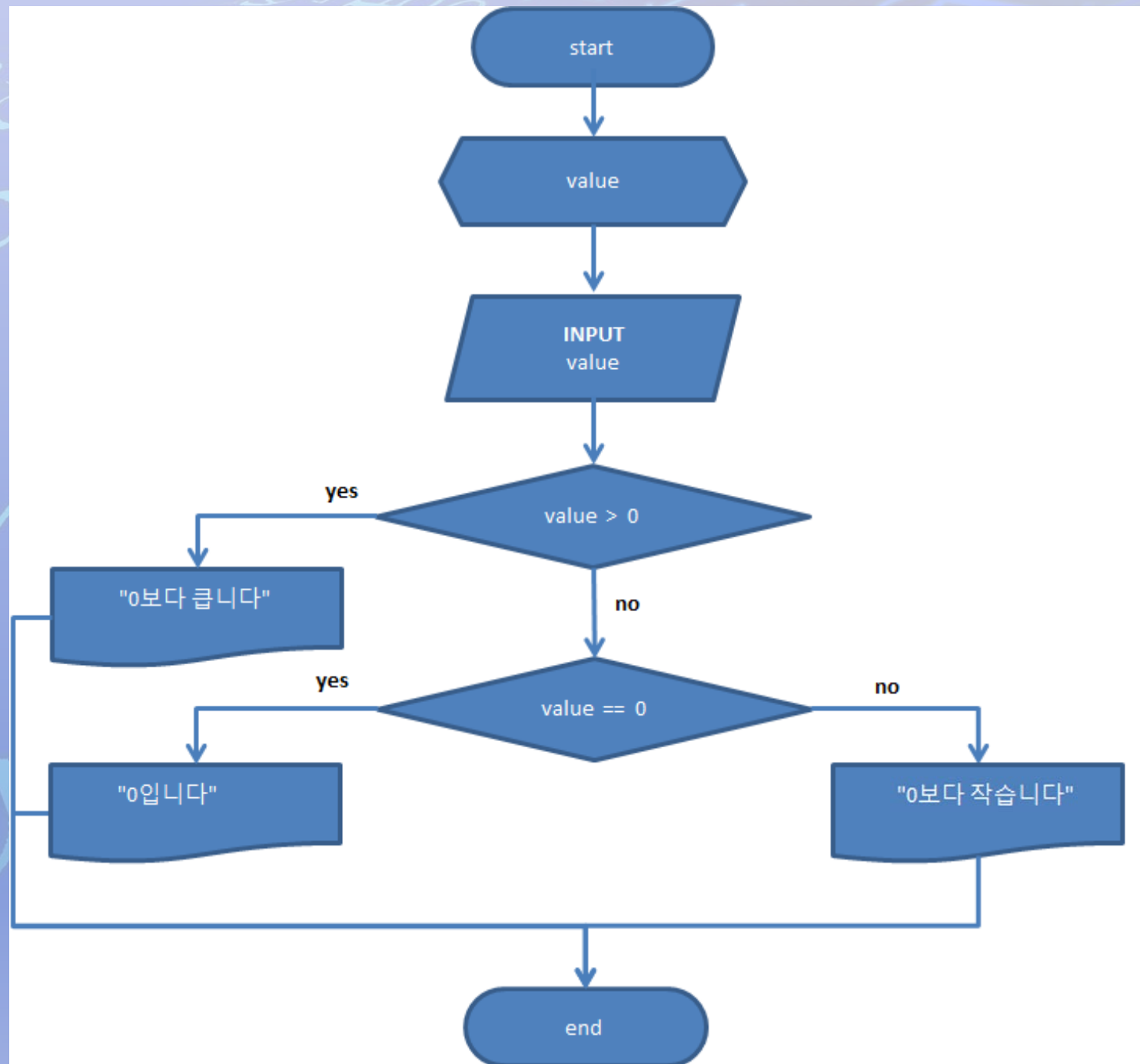
# 확인 문제

- 사용자로부터 하나의 정수 값을 입력 받아 변수 value에 저장한 후, 그 값이 0보다 크면 "value는 0보다 큼니다", 0과 같으면 "value는 0입니다", 0보다 작으면 "value는 0보다 작습니다"라고 출력하는 순서도를 다중선택 if문을 사용하여 작성하여라.

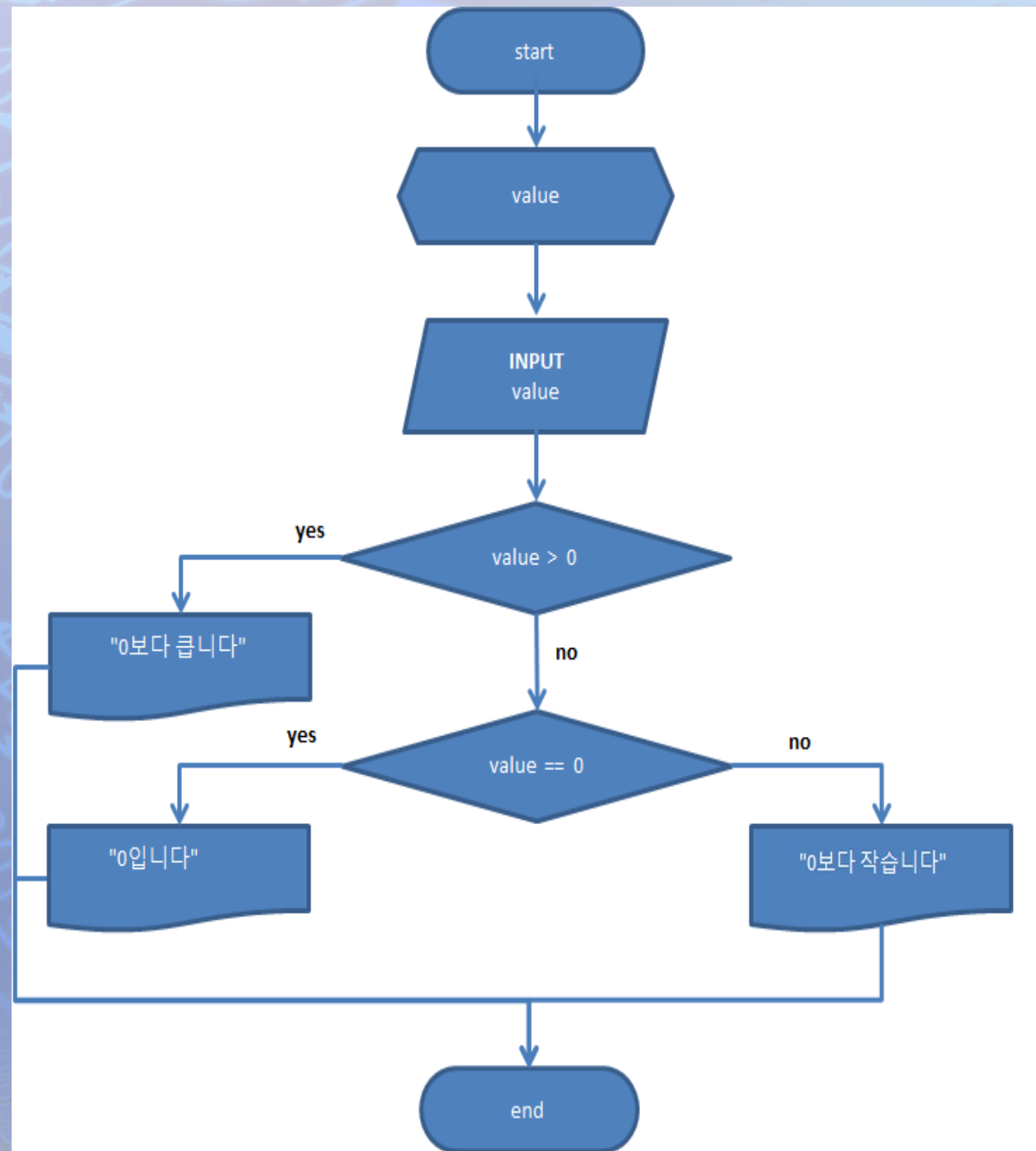
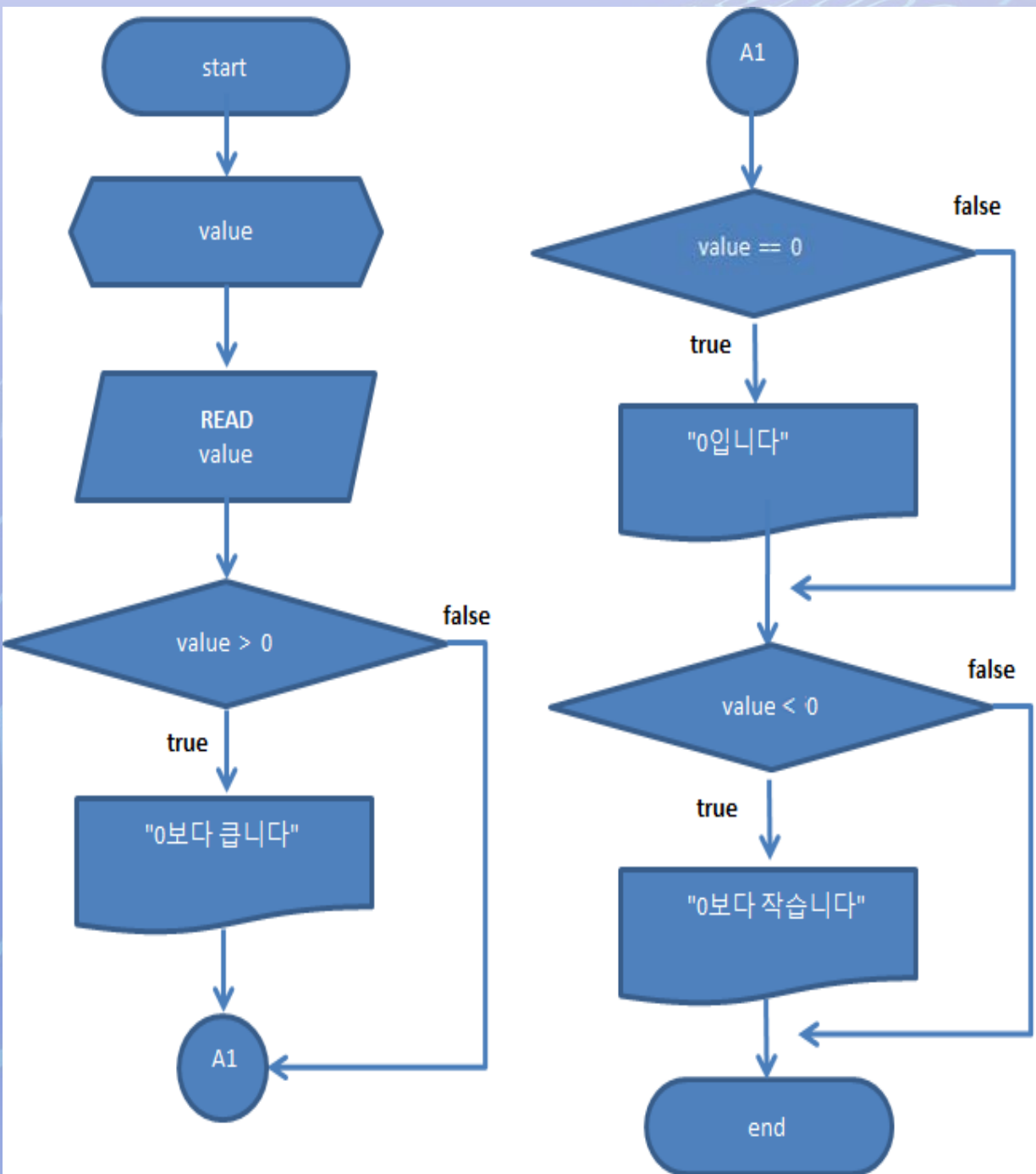
## [문제해결논리]

"value는 0보다 큼니다", "value는 0입니다", "value는 0보다 작습니다"를 다중선택 if문의 조건들로 사용한다.









```
ex23_prompt_4_y_z - 메모장
파일(F) 편집(E) 서식(O) 보기(V) 도움말(H)

<html>
<head> <title> 임의의 정수 값 판단 </title> </head>
<body>
  <script type = "text/javascript">

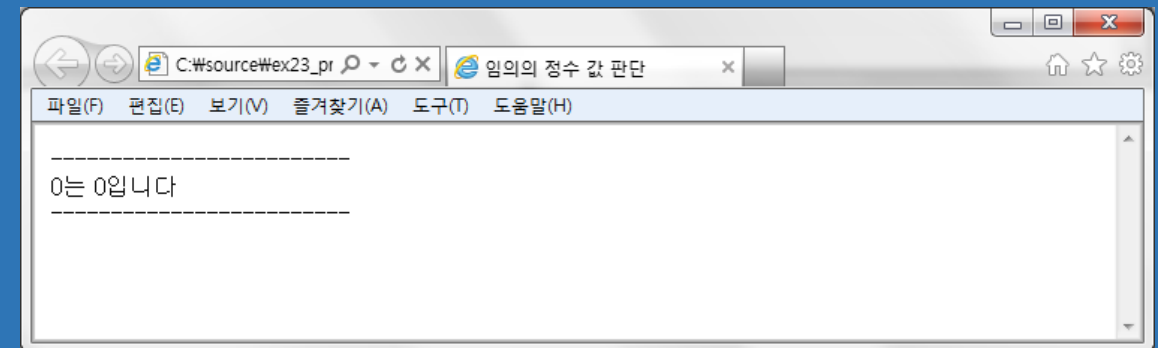
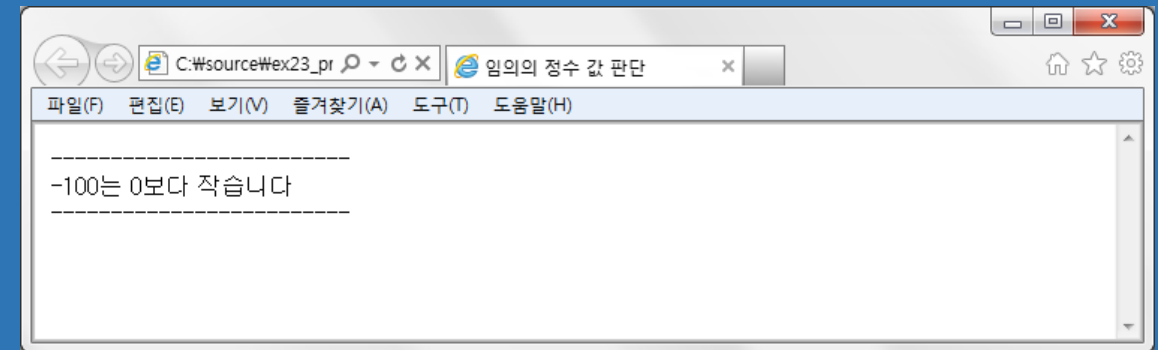
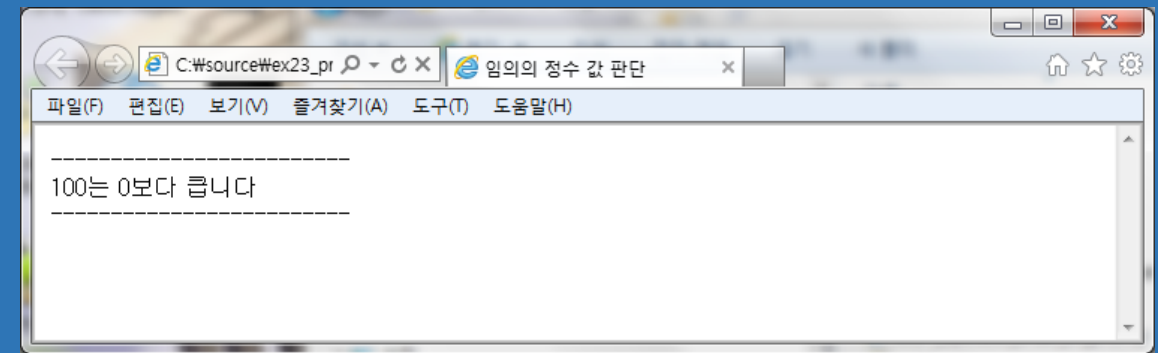
    var value;

    document.write("-----<br>");

    value = prompt("임의의 정수값을 입력하세요", "0");

    if(value > 0) document.write(value + "는 0보다 큼니다" + "<br>");
    else if(value == 0) document.write(value + "는 0입니다" + "<br>");
    else document.write(value + "는 0보다 작습니다" + "<br>");

    document.write("-----<br>");
  </script>
</body>
</html>
```



## [정리]

단일선택 if문 및 양쪽선택 if문을 이용하여 작성한 것과 조금 다르다는 것을 알 수 있다. 이런 유형의 문제는 다중선택 if문을 이용하여 작성하는 것이 문제 해결을 쉽게 만들며, 프로그램도 간결해진다.

# 선택 논리를 위한 세 가지 if문

- 어떤 경우에 어떤 if문을 사용하는지에 대한 기준이 있는가?
  - 판단해야 되는 조건에 의한 선택이 1개라면 단일선택 if문 사용
  - 조건에 의한 선택이 2개이면 양쪽선택 if문 사용
  - 조건에 의한 선택이 3개 이상이면 다중선택 if문 사용
  - 문제를 해결하는데 쉽고 바람직



# 일상 생활에서의 예

일상생활의 예	if문으로 표현
지갑에 돈이 5,000원 이상 있는 경우, 택시를 타고 서점에 가서 책을 구매한 후, 집으로 돌아와 휴식을 취한다. 그렇지 않으면 그냥 집으로 돌아와 휴식만 취한다.	단일선택 if문 ← 조건에 의한 서점에 가는 선택이 1가지  if(돈 >= 5000) { 택시를 타고 서점에 간다; 책을 구매한다; }  집으로 돌아온다;  휴식을 취한다;

# 일상 생활에서의 예

일상생활의 예	if문으로 표현
지갑에 돈이 5,000원 이상 있는 경우, 택시를 타고 서점에 가서 책을 구매한다. 그렇지 않으면 걸어서 서점에 가서 책을 구매한다. 책을 구매한 후에는 집으로 돌아와 책을 읽는다.	양쪽선택 if문 ← 조건에 의한 서점에 가는 선택이 2가지 if(돈 >= 5000) { 택시를 타고 서점에 간다; } else { 걸어서 서점에 간다; } 책을 구매한다; 집으로 돌아온다; 책을 읽는다.

# 일상 생활에서의 예

일상생활의 예	if문으로 표현
지갑에 돈이 5,000원 이상 있는 경우, 택시를 타고 서점에 가서 책을 구매한다. 그렇지 않고 지갑에 돈이 5,000원 미만 있는 경우, 버스를 타고 서점에 가서 책을 구매한다. 돈이 없다면 걸어서 서점에 간다. 책을 구매한 후 집으로 돌아와 책을 읽는다.	다중선택 if문 ← 조건에 의한 서점에 가는 선택이 3가지  if(돈 >= 5000) { 택시를 타고 서점에 간다; } else if(0 < 돈 < 5000){ 버스로 서점에 간다; } else { 걸어서 서점에 간다; }  책을 구매한다;  집으로 돌아온다;  책을 읽는다.



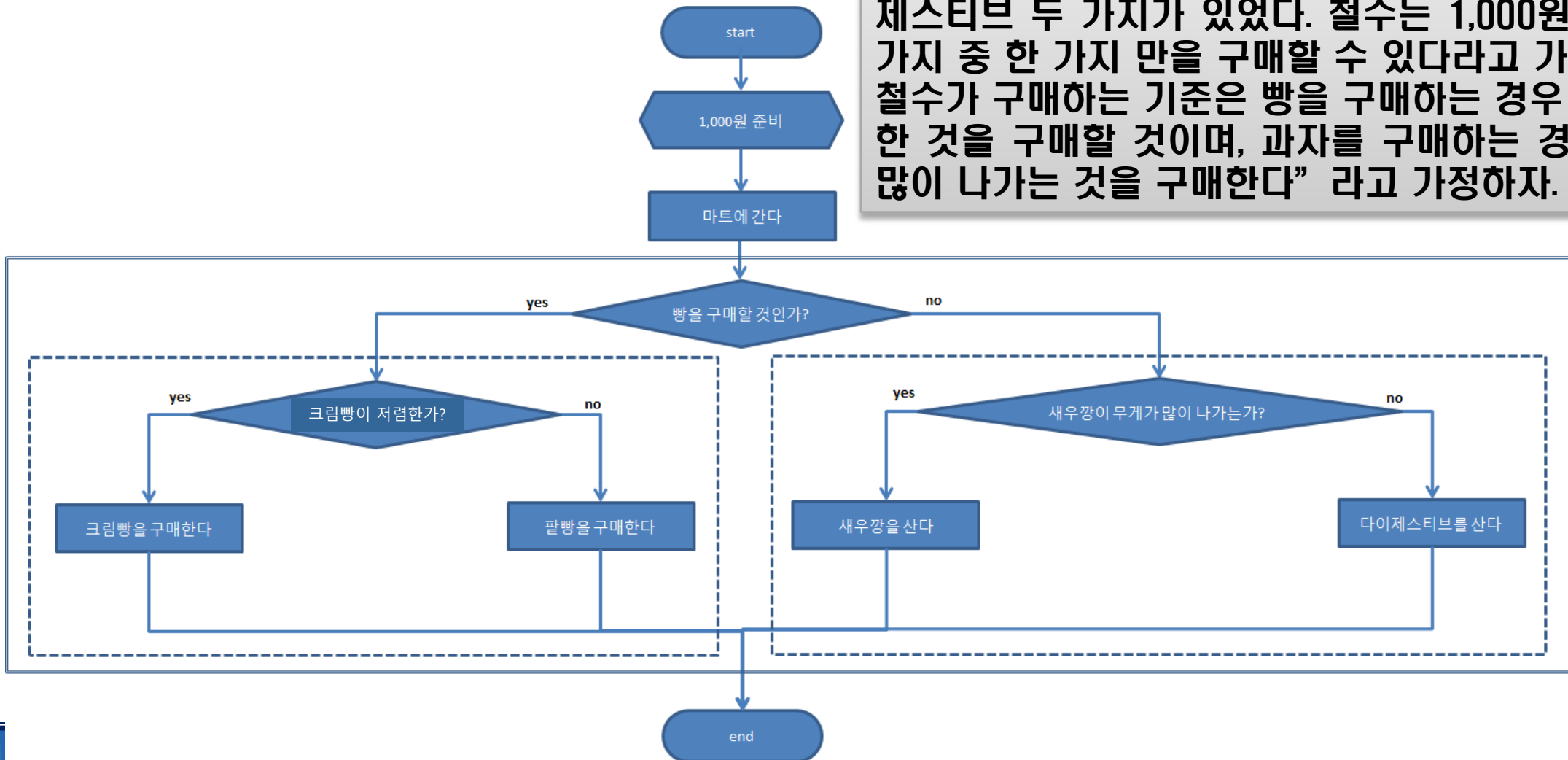
## 2.4 if문의 중첩

### ■ if문의 중첩

- if문 내에 1개 이상의 if문이 들어있는 구조
- 예
  - 단일선택 if문의 내부 어떤 위치에든 관계없이 양자선택 if문 또는 다중선택 if문이 들어갈 수 있음
  - 양자선택 if문의 내부 어떤 위치에든 관계없이 단일선택 if문 또는 다중선택 if문이 들어갈 수 있음
  - 다중선택 if문의 내부 어떤 위치에든 관계없이 단일선택 if문 또는 양자선택 if문이 들어갈 수 있음
- 일상생활에서도 많이 찾아볼 수 있음

## 2.4 if문의 중첩

### ■ 일상생활에서의 예



철수는 오늘 용돈 1,000원을 받았다. 철수는 마트에 가서 빵과 과자 둘 중의 하나를 구매할 예정이다. 잠시 후, 철수는 마트에 간다. 마트를 둘러보니 빵과 과자가 보인다. 빵은 크림빵과 팥빵 두 가지가 있고, 과자도 새우깡, 다이제스티브 두 가지가 있었다. 철수는 1,000원으로 이들 4가지 중 한 가지 만을 구매할 수 있다고 가정하자. 또한 철수가 구매하는 기준은 빵을 구매하는 경우 가격이 저렴한 것을 구매할 것이며, 과자를 구매하는 경우는 무게가 많이 나가는 것을 구매한다” 라고 가정하자.

## 2.4 if문의 중첩

### ■ 일상생활에서의 예

조건 부분만을 if문으로 간략하게 표현

```
if(빵을 구매할 것인가?) {  
    if(크림빵이 저렴한가?) { 크림빵을 구매한다; }  
    else { 팔빵을 구매한다; }  
}  
else  
{  
    if(새우깡이 무게가 많이 나가는가?) { 새우깡을 구매한다; }  
    else { 다이제스티브를 구매한다; }  
}
```



# 선택 논리를 위한 제어문

# if와 else의 매칭 문제

else 절은 가장 가까운  
if절과 매치된다.

```
if(score >= 80)
    if( score >= 90)
        printf("당신의 학점은 A입니다\n");
    else
        printf("당신의 학점은 B입니다\n");
```

```
if( score >= 80 )
{
    if( score >= 90 )
        printf("당신의 학점은 A입니다.\n");
}
else
    printf("당신의 학점은 A나 B가 아닙니다.\n");
```

만약 다른 if절과 else 절  
을 매치시키려면 중괄호  
를 사용하여 블록으로 묶  
는다.

# else 주의

## ■ 잘못된 들여쓰기

- else의 들여쓰기는 첫 번째 if(num1) 문의 else 문으로 처리하였으나 실제로는 두 번째 if(num2)의 else 문
- 에러는 없으나 이해가 어려움

```
if (num1)
    if (num2)
        printf("두 수 모두 0이 아닙니다.\n");
else
    printf("한 수는 0이고 다른 한 수는 0이 아닙니다.");
```

이 문장은 num1 이 참(0이 아니고)이고 num2가 거짓(0이면)이면 실행된다. ↵



# else 주의

- 수정 : 다음과 같이 들여쓰기

```
if (num1)
    if (num2)
        printf("두 수 모두 0이 아닙니다.\n");
    else
        printf("한 수는 0이고 다른 한 수는 0이 아닙니다. );
```

```
if (num1) {
    if (num2)
        printf("두 수 모두 0이 아닙니다.\n");
    else
        printf("한 수는 0이고 다른 한 수는 0이 아닙니다. );
}
```

# else if 사용

## ■ 예제

- 입력한 정수가 1에서부터 45까지의 정수인지를 검사하여, 1보다 작은 수, 45보다 큰 수, 아니면 1에서부터 45까지 유효한 수임을 출력하는 프로그램

"D:\src\08\elseifvalid\Debug\elseifvalid.exe"

Enter a number between 1 and 45. -->30  
The number 30 is valid.

"D:\src\08\elseifvalid\Debug\elseifvalid.exe"

Enter a number between 1 and 45. -->46  
Number is above 45.  
The number 46 is not valid.

```
int main(void)
{
    int number = 0, valid = 0;

    printf("Enter a number between 1 and 45. -->");
    scanf("%d", &number);

    if ( number < 1 ) {
        printf("Number is below 1.\n");
        valid = 0;
    } else if( number > 45 ) {
        printf("Number is above 45.\n");
        valid = 0;
    } else
        valid = 1;

    if ( valid == 1 ) {
        printf("The number %d is valid.\n", number );
    } else {
        printf("The number %d is not valid.\n", number );
    }

    return 0;
}
```

# else if 사용

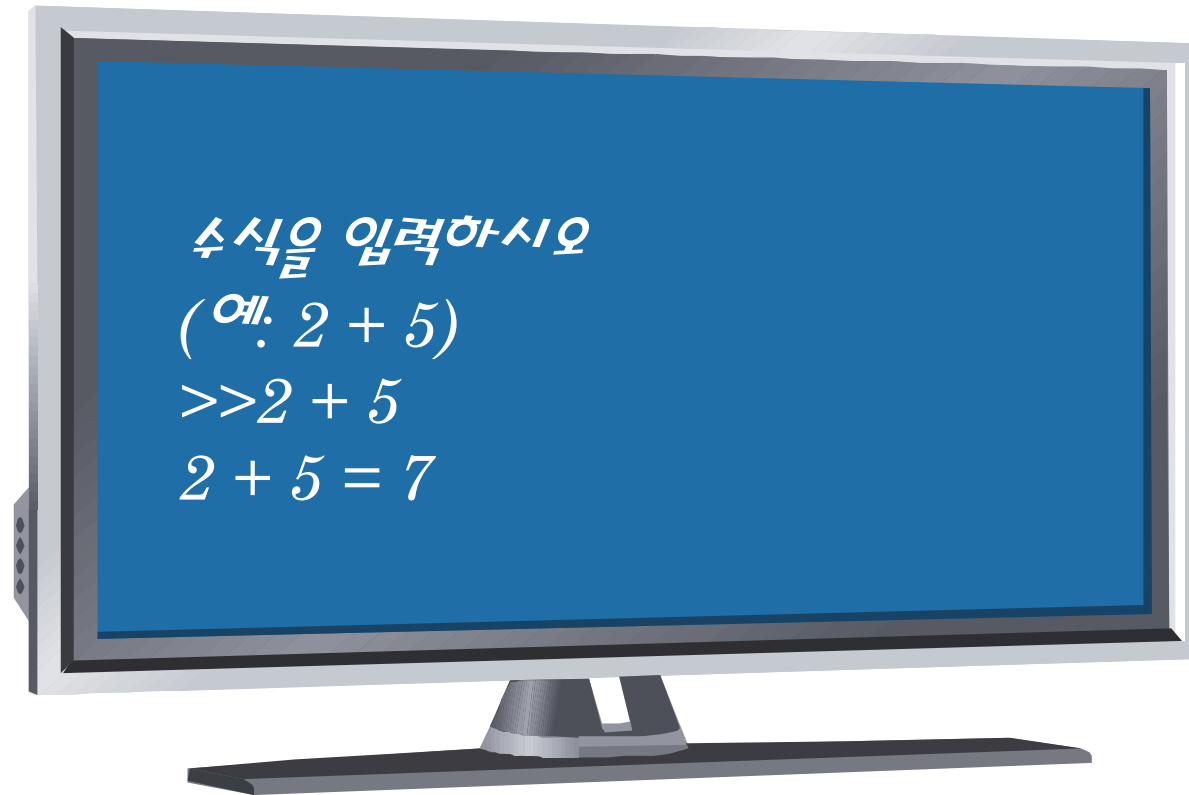
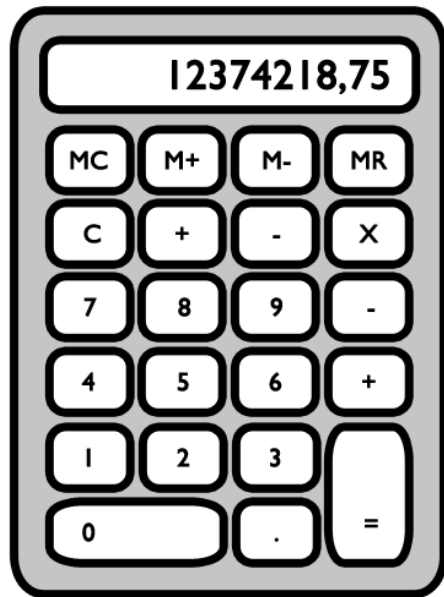
## ■ 수의 검사

- 입력한 정수가 1에서부터 45까지의 정수인지를 검사하는 프로그램 모듈
  - 정수를 1보다 작은 수, 45보다 큰 수, 그리고 1에서 45 사이의 수로 구분하는 모듈을 작성
  - 다음과 같이 else if 문을 이용하면 쉽게 정수를 세 부분으로 나눌 수 있다.

```
if ( number < 1 ) {  
    printf("Number is below 1.\n");  
    valid = 0;  
}  
else if ( number > 45 ) {  
    printf("Number is above 45\n");  
    valid = 0;  
}  
else  
    valid = 1;
```

여기서 조건식은 else에 의하여 (number >= 1 && number > 45)을 의미한다.

# 산술 계산기





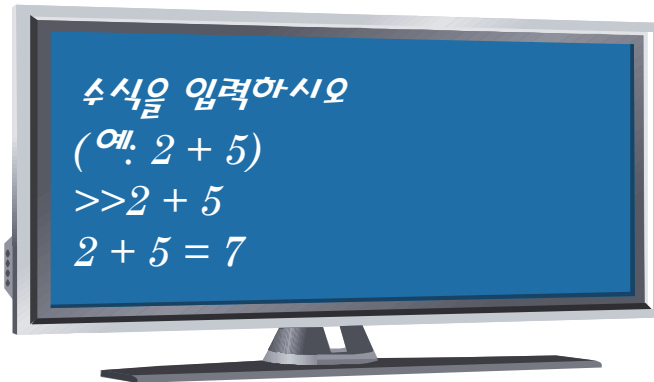
# 산술 계산기

...

```
int main(void)
{
    char op;
    int x, y, result;

    printf("수식을 입력하십시오");
    printf("(예: 2 + 5) ");
    printf(">>");

    scanf("%d %c %d", &x, &op, &y);
```



```
if( op == '+' )
    result = x + y;
else if( op == '-' )
    result = x - y;
else if( op == '*' )
    result = x * y;
else if( op == '/' )
    result = x / y;
else if( op == '%' )
    result = x % y;
else
    printf("지원되지 않는 연산자입니다. ");

printf("%d %c %d = %d ", x, op, y, result);
return 0;
}
```

# 예제

- n의 값이 각각 -1, 0, 5인 경우에 다음의 코드에 의하여 생성되는 출력은 무엇인가?

```
if( n == 0 )  
    printf("A");  
else if( n > 3 )  
    printf("B");  
else  
    printf("C");
```

[출력] C A B

# 예제

- 컵의 사이즈를 받아서 100ml미만은 small, 100ml이상 200ml 미만은 medium, 200ml 이상은 large라고 출력하는 연속적인 if-else 문을 작성하시오.

```
if(cup < 100) printf("small");  
else if(cup<200) printf("medium");  
else printf("large");
```

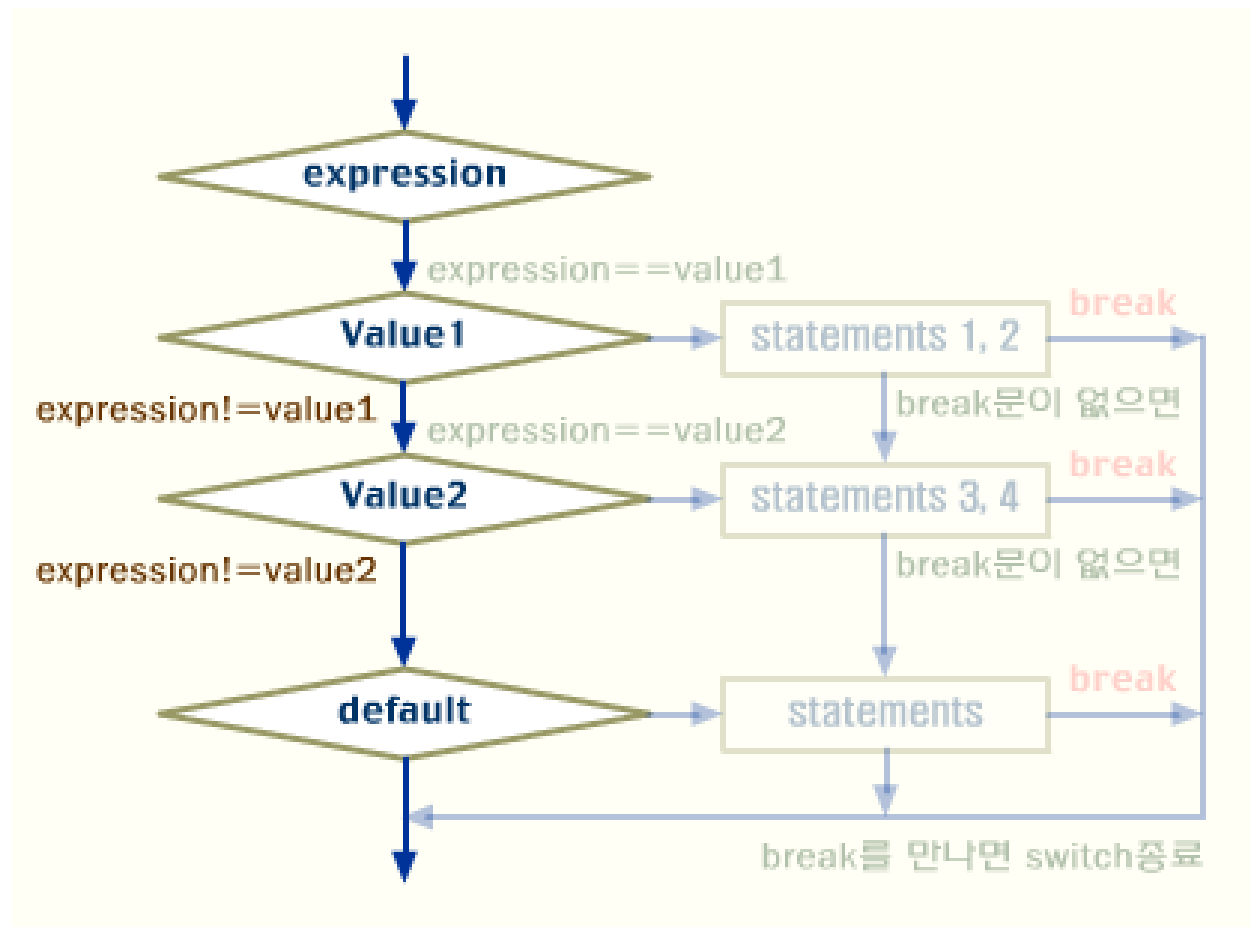
# switch문

- switch문은 if문처럼 조건식이 true일 경우에 블록 내부의 실행문을 실행하는 것이 아니라, 조건식이 어떤 값을 갖느냐에 따라 실행문이 선택됨.
- if문의 단점
  - 조건식의 결과가 true, false 두 가지밖에 없기 때문에, 경우의 수가 많아 질수록 else if를 반복적으로 추가해야 함.
  - 따라서 코드가 복잡해짐
- switch문은 조건식에 따라 실행문이 결정됨
  - 같은 기능의 if문보다 코드가 간결함
- case 끝에 break가 붙는 이유
  - 다음 case를 실행하지 말고 switch문을 빠져나가기 위해서.
  - break가 없다면 다음 case가 연달아 실행되어 case값과 상관없이 실행됨.



# switch문

```
switch ( expression ) {  
    case value1:  
        statement1;  
        statement2;  
        .....  
        break;  
  
    case value2:  
        statement3;  
        statement4;  
        .....  
        break;  
  
    ...  
    default:  
        statement  
        .....  
        .....  
        break;  
}
```



# switch문 이용 주의사항

## ■ 키워드

- switch 문에서 switch, case, break, default는 키워드이며, expression과 value는 반드시 정수나 정수 수식이어야 한다.
- 또한 case 다음의 value 값은 중복되어서는 안 된다.

## ■ switch 문의 정확한 실행 순서

- switch 문의 표현식을 평가
- 위 표현식에서 계산된 값과 일치하는 상수 값을 갖는 case의 값을 위에서부터 찾는다. 일치된 case 값을 만나면 case 내부의 문장을 실행
- break를 만나면 switch 문을 종료, 또는 switch 몸체의 마지막 문장을 수행하면 switch 문을 종료
- 일치된 case 값을 만나지 못하여 default를 만나면 default 내부의 문장을 실행

# switch문

- 여러가지 경우 중에서 하나를 선택하는데 사용

**switch**(조건식)

{

**case** *c1*:  
문장 1;  
**break**;

조건식의 값이 *c1*이면 실행된다.

**case** *c2*:  
문장 2;  
**break**;

조건식의 값이 *c2*이면 실행된다.

...

...

**case** *cn*:  
문장 *n*;  
**break**;

**default**:  
문장 *d*;  
**break**;

일치하는 값이 없으면 실행된다.

}

# 예제

## ■ 간단한 계산기 프로그램을 switch 문을 이용하여 작성

- 두 실수의 더하기, 빼기, 곱하기, 나누기를 할 수 있는 계산기 프로그램을 작성
- 표준입력으로부터 3 \* 4 와 같이 계산식을 입력 받아 그 결과를 출력하는 프로그램

## ■ 구현

- 변수 operator는 문자형 char 자료형으로
  - 연산자 문자인 \*, /, +, - 중의 하나를 저장한 변수

실행에서 default를 만난다는 것은 변수 operator 값이 네 개의 연산자 case와 일치하지 않음을 의미하므로, 이 문장은 잘못된 연산자를 입력한 경우 실행된다.

```
switch( operator ) {  
    case '*' :  
        result = number1 * number2;  
        break;  
  
    case '/' :  
        result = number1 / number2;  
        break;  
  
    case '+' :  
        result = number1 + number2;  
        break;  
  
    case '-' :  
        result = number1 - number2;  
        break;  
  
    default :  
        invalid_operator = 1;  
}
```



# 예제

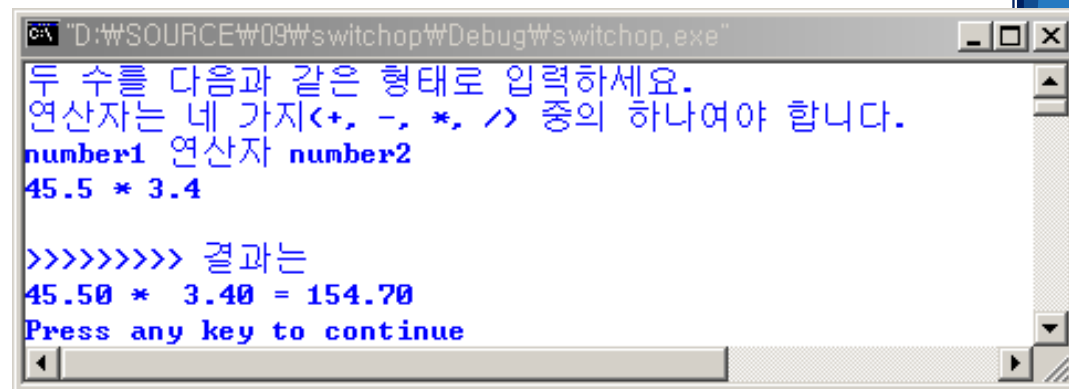
```
int main(void)
{
    int  invalid_operator = 0;
    char operator;
    float number1, number2, result;

    printf("두 수를 다음과 같은 형태로 입력하세요.\n");
    printf("연산자는 네 가지(+, -, *, /) 중의 하나여야 합니다.\n");
    printf("number1 연산자 number2\n");
    scanf("%f %c %f", &number1, &operator, &number2);

    switch( operator ) {
        case '*' :
            result = number1 * number2;
            break;
        case '/' :
            result = number1 / number2;
            break;
        case '+' :
            result = number1 + number2;
            break;
        case '-' :
            result = number1 - number2;
            break;
        default :
            invalid_operator = 1;
    }

    switch( invalid_operator ) {
        case 1 : printf("연산자가 잘못 입력되었습니다.\n");
                break;
        default :
            printf("\n>>>>>>>> 결과는 \n");
            printf("%5.2f %c %5.2f = %5.2f\n", number1, operator, number2, result);
            break;
    }

    return 0;
}
```



```
D:\SOURCE\09\switchop\Debug\switchop.exe
두 수를 다음과 같은 형태로 입력하세요.
연산자는 네 가지(+, -, *, /) 중의 하나여야 합니다.
number1 연산자 number2
45.5 * 3.4

>>>>>>>> 결과는
45.50 * 3.40 = 154.70
Press any key to continue
```

# break 역할

## ■ 중요한 break

- switch 문에서 break 문을 만나면 무조건 switch 문을 종료
- 그러나 switch 문의 case 문 내부에 break 문이 없다면
  - 일치하는 case 문을 실행하고,
  - 계속해서 break 문을 만나기 전까지 무조건 다음 case 문 내부의 문장을 실행

## ■ 구문 예

- 아래 구문에서 변수 month의 값이 1이거나 3, 10, 12 중의 하나이면 case 12:의 내부 문장 두 개를 실행하고, break 문장을 만나 switch 문이 종료

```
switch (month) {  
    case 1:   
    case 3:  
    case 10:  
    case 12:  
        maxDay = 31;  
        validinput = 1;  
        break;  
  
    case 4:  
    case 6:  
    case 9:  
    case 11:  
        maxDay = 30;  
        validinput = 1;  
        break;  
}
```

month가 1인 경우, break 문이 없으므로 case 3, 10, 12를 거쳐 두 문장을 실행하고 switch 문을 종료한다. ↵

위와 마찬가지로 month가 4, 또는 6, 9, 11 중의 하나이면 이 두 문장을 실행한다. ↵

# switch문 → if문

## ■ 프로그램 목적

- switch 문을 이용한 사칙연산 문제에서 switch 문 대신에 if 문을 이용하는 프로그램
- 두 실수의 더하기, 빼기, 곱하기, 나누기를 할 수 있는 계산기 프로그램을 작성
- 표준입력으로부터  $3 * 4$  와 같이 계산식을 입력 받아 그 결과를 출력하는 프로그램

## ■ 구현

- 위 프로그램에서 유효한 계산식을 입력할 수 있도록 사용자에게 알리는 출력문을 보내고, 계산식을 입력 받아 연산자의 구분에 따라 적절한 계산을 수행하는 프로그램을 구현

# switch문 → if문 : 소스와 결과

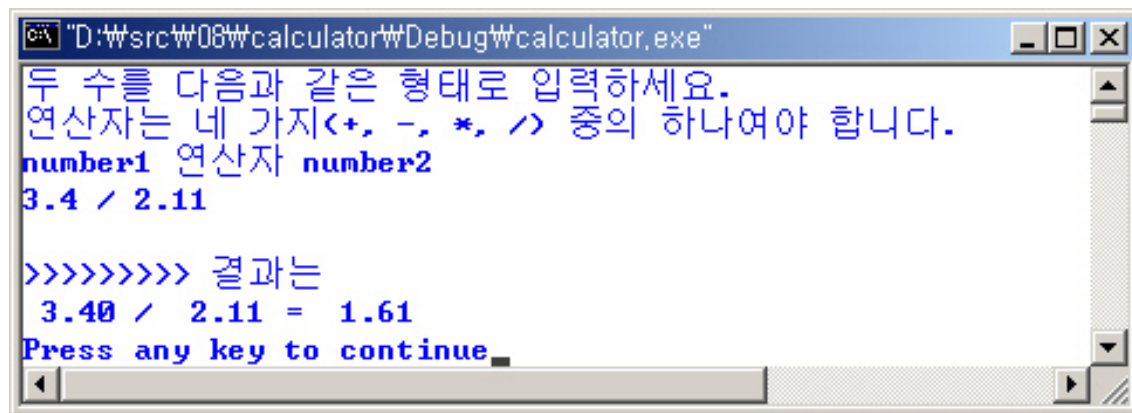
```
int main(void)
{
    int    invalid_operator = 0;
    char    operator;
    float   number1, number2, result;

    printf("두 수를 다음과 같은 형태로 입력하세요.\n");
    printf("연산자는 네 가지(+, -, *, /) 중의 하나여야 합니다.\n");
    printf("number1 연산자 number2\n");
    scanf("%f %c %f", &number1, &operator, &number2);

    if(operator == '*')
        result = number1 * number2;
    else if(operator == '/')
        result = number1 / number2;
    else if(operator == '+')
        result = number1 + number2;
    else if(operator == '-')
        result = number1 - number2;
    else
        invalid_operator = 1;

    if( invalid_operator != 1 ) {
        printf("\n>>>>>>>> 결과는 \n");
        printf("%5.2f %c %5.2f = %5.2f\n", number1, operator, number2, result);
    } else
        printf("연산자가 잘못 입력되었습니다.\n");

    return 0;
}
```



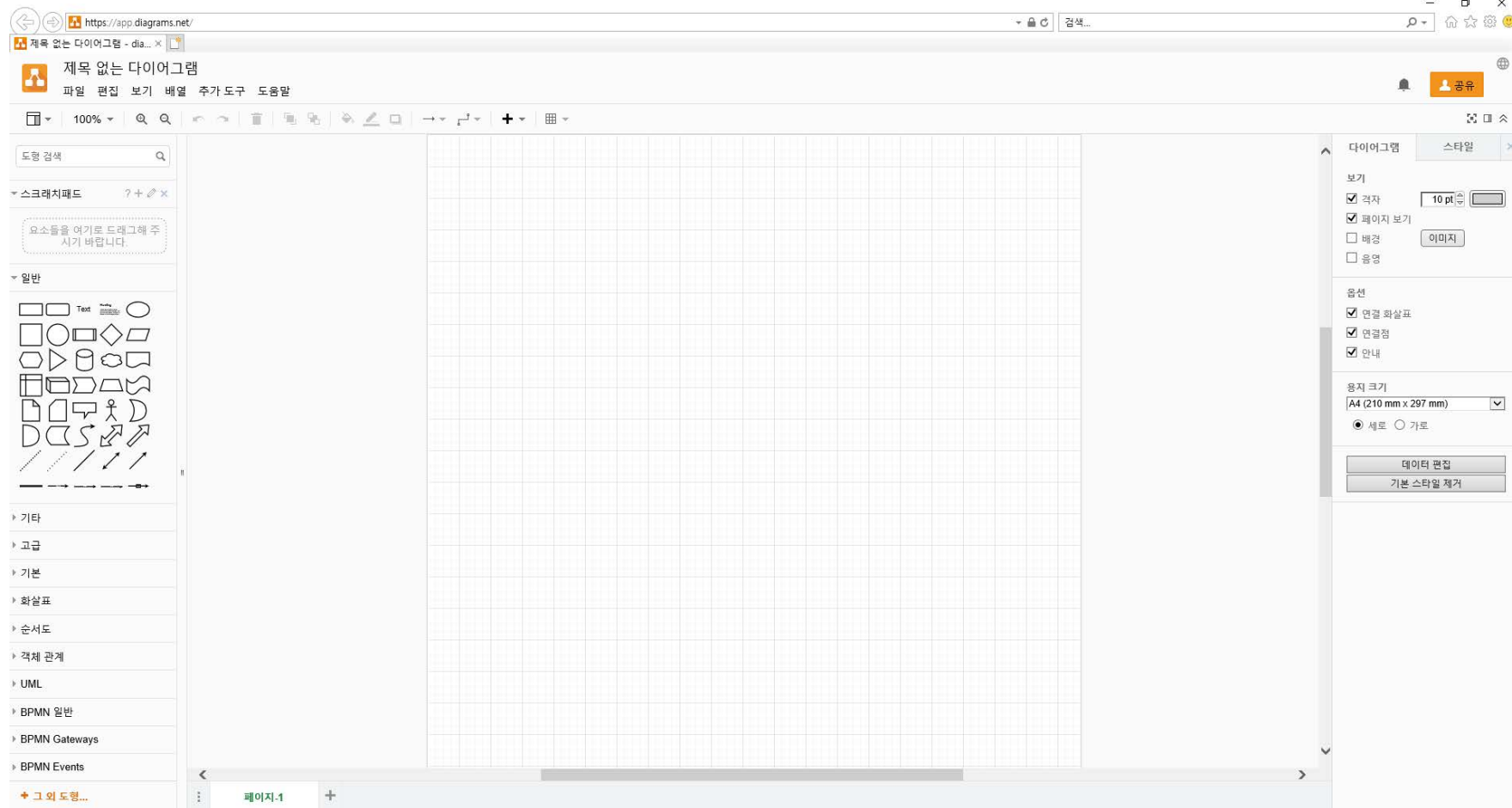
```
D:\src\W08\calculator\Debug\calculator.exe
두 수를 다음과 같은 형태로 입력하세요.
연산자는 네 가지(+, -, *, /) 중의 하나여야 합니다.
number1 연산자 number2
3.4 / 2.11

>>>>>>>> 결과는
 3.40 / 2.11 = 1.61
Press any key to continue
```



# 순서도 그리기

- <https://app.diagrams.net>



## ■ 참고 자료

- 최명복, 컴퓨터 프로그래밍 문제해결기법, 생능출판사, 2014.



다음 시간

8주차 : 중간평가