
C 프로그래밍 및 실습

5. 조건문

세종대학교

목차

- 1) 조건문 개요
- 2) if 문
- 3) if-else 문
- 4) 다중 if 문
- 5) switch 문

1) 조건문 개요

▪ 제어문

- C 언어는 순차처리언어로, 특별한 지정이 없으면, 소스 코드 첫 줄부터 차례대로 처리
- 그러나 문제 해결 위해 처리 흐름 제어 필요 → 제어문
- C언어에서는 [조건문\(5장\)](#), 반복문(6장)과 같은 제어문 제공

▪ 제어문 종류

분류	종류
조건문	if 문 , if-else 문 , switch 문
반복문	for 문, while 문, do-while 문
기타	break 문 , continue 문, goto 문, return 문

1) 조건문 개요

- 조건문 예시

- 입력된 정수의 절댓값을 출력하는 프로그램
 - ✓ 입력된 정수가 음수인 경우, 부호 바꾸어 주는 작업 필요
 - ✓ 하지만, 양수인 경우, 필요 없음
 - ✓ C 언어 문장으로 표현하면?

$a < 0$ 이면 $a = -a;$ ➔ **if**($a < 0$) $a = -a;$

- 조건문

- 조건에 따라 실행되는 코드가 달라지게 할 때 사용하는 제어문
- if 문, if-else 문, switch 문

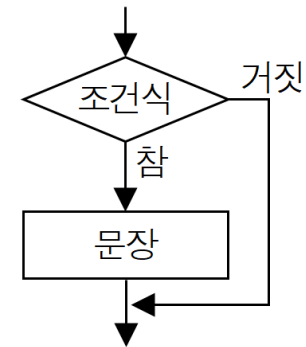
2) if 문

▪ if 문

- 조건식이 참이면 중괄호로 둘러 쌓인 블록 안의 문장들이 실행된다.

- 구문

```
if( 조건식 )  
{  
    문장;  ⇨ 조건식이 참일 때만 실행  
} ⇨ 블록 안의 문장이 하나인 경우  
      중괄호 생략 가능
```



- 예) x의 값이 0 보다 큰 경우에만 "양수입니다" 출력

```
if( x > 0 )  
{  
    printf("양수입니다.\n");  
}
```

2) if 문

- 다음 각 문장에 해당하는 if 문을 작성하시오 (각 문제는 독립)
 1. 변수 x가 0보다 작거나 같으면, x의 값 1 증가
 2. 변수 a가 -10보다 크고 10보다 작으면, x에 0 대입
 3. 변수 num이 0이면, "0입니다" 를 화면에 출력

2) if 문

- if 문의 조건식이 참일 경우 **실행할 문장이 하나**라면 중괄호 { }는 생략 가능

```
if(x > 0)
{
    printf("양수입니다.\n");
}
```

=

```
if(x > 0)
    printf("양수입니다.\n");
```

실행 예시 1 (x가 9인 경우)

양수입니다.

실행 예시 2 (x가 -9인 경우)

2) if 문

■ 기본 예제

- 정수 한 개를 입력 받고 절대값을 출력하는 프로그램을 작성하시오.
 - ✓ 어떻게 프로그램을 만들지 생각해 보자.
 - ① 사용자 정수 입력
 - ② 정수 검사(절대값을 계산하기 위해 무엇을 할 것인가?)
 - ③ 절대값 출력

실행 예시 1

정수를 입력하세요: 20
절대값 = 20

실행 예시 2

정수를 입력하세요: -20
절대값 = 20

2) if 문

- 답 예시

- 아래 코드 외에 다른 방식으로 작성하는 것도 가능

```
#include <stdio.h>
int main(){
    int x = 0;

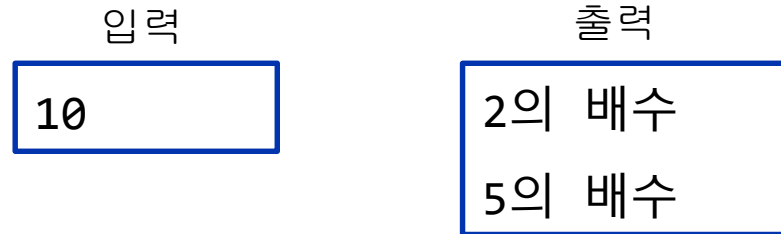
    printf("정수를 입력하세요: ");
    scanf("%d", &x);

    if( x < 0 )
        x = -x;
    printf("절대값 = %d\n", x);

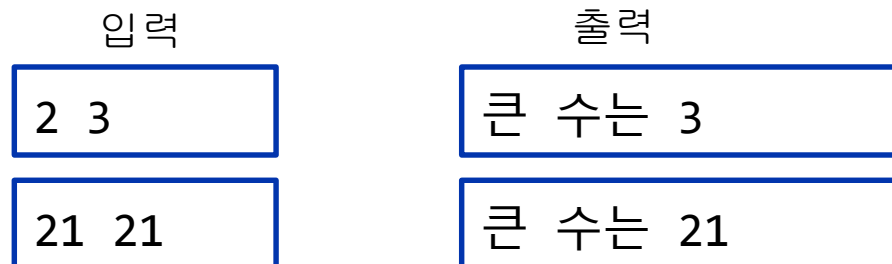
    return 0;
}
```

2) if 문

- **[예제 5.1]** 하나의 양의 정수를 입력 받아서, 2의 배수, 3의 배수, 5의 배수인지 조사하는 프로그램을 작성하시오. (힌트: if 문 3개 사용)



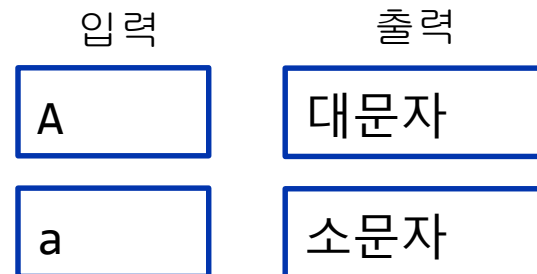
- **[예제 5.2]** 두 개의 정수를 입력 받아서, **큰 수**를 출력하는 프로그램을 작성하시오. (나중에 배울 if-else 문이 더 적절하지만, 여기서는 if 문 2 개를 사용해서 작성하자.)



2) if 문

- **[예제 5.3]** 영어 소문자가 입력되면 "소문자"를 출력하고, 대문자가 입력되면 "대문자"라고 출력하는 코드를 작성하시오.
 - 1) 아래의 아스키 문자표를 이용하여 해결하시오.
 - 2) 아스키 문자표를 이용하지 않고 해결해 보시오.

영어 문자	십진수
A	65
B	66
Z	90
a	97
b	98
z	122



2) if 문

■ 들여쓰기

- 일반적으로 if문의 조건에 의해 영향을 받는 문장들은 쉽게 구별하기 위해 if 보다 더 들여 씀 (보통 tab 키 사용)

(들여쓰기) `if(x > 0)`
`----> printf("%d는 양수\n", x);`

- 들여쓰기는 **가독성**을 위한 것 일 뿐, 구문에는 전혀 영향 없음
✓ 아래에서 첫 번째 printf만 if문에 의해 영향을 받음

```
int x = -2;
if(x > 0)
    printf("%d는 양수\n", x);
    printf("%d는 0보다 큼\n", x);
```

-2는 0보다 큼

- 위 코드를 구문에 맞게 들여쓰기를 하면?
- 위 코드를 의미에 맞게 수정하려면? (즉, $x > 0$ 인경우에만, 두 printf 문장 수행)

2) if 문

- if 문 사용 시 주의 사항

- if의 조건식 뒤에 바로 세미콜론 ; 은 쓰지 않는다.
 - ✓ 세미콜론만 써도 아무런 일도 하지 않는 하나의 문장으로 인식

```
int x = -2;  
if(x > 0) ;  
    printf("%d는 양수\n", x);
```

-2는 양수

||

```
int x = -2;  
if(x > 0)  
    ;           ⇒ 빈 문장  
printf("%d는 양수\n", x);
```

2) if 문

- if 문 사용 시 자주하는 실수

- 조건식에서 비교연산자 `==` 가 아닌 대입 연산자 `=` 를 쓰는 경우

```
int x = -2;  
if( x = 9 )           ⇒ 대입 연산: 참  
    printf("x는 %d\n",x);
```

✓ 위 코드는 컴파일 오류가 발생하지 않음 (즉, 문법적 오류 없음)

✓ 실행 결과는? x는 9

✓ 위 코드의 의미: **x에 9를 대입하고, x의 값을 조건식으로 사용**
(C언어에서 0이 아니면 모두 참)

✓ if문의 조건식을 **x = 1** 또는 **x = 0** 으로 바꿔서 실행해보자.

목차

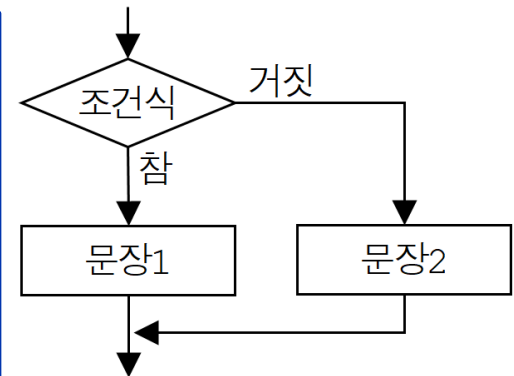
- 1) 조건문 개요
- 2) if 문
- 3) **if-else 문**
- 4) 다중 if 문
- 5) switch 문

3) if-else 문

- if-else 문

- if의 조건식이 참이면 조건식 바로 다음의 블록이 실행되고, 조건식이 거짓이면 else 문 다음의 블록이 실행된다.
- 구문:

```
if( 조건식 ) {  
    문장1; ⇨ 조건식이 참일 때 실행  
}  
else {  
    문장2; ⇨ 조건식이 거짓일 때 실행  
}
```



- ✓ 각 블록 안의 문장이 하나인 경우, 해당 블록의 종괄호 생략 가능

```
if(x >= 0)  
    printf("0 또는 양수\n");  
else  
    printf("음수\n");
```


3) if-else 문

- 다음 각 문장에 해당하는 if-else 문을 작성하시오 (각 문제는 독립)
 1. 변수 x가 0보다 작거나 같으면, x의 값 1 증가
그렇지 않으면, x의 값 1 감소
 2. 변수 a가 -10보다 크고 10보다 작으면, x에 0 대입
그렇지 않으면, x에 1을 대입
 3. 변수 num이 0이면, "0입니다" 를 화면에 출력
그렇지 않으면 "0이 아니다"를 화면에 출력

3) if-else 문

- **[예제 5.4]** 정수 두 개를 입력받아 큰 수를 출력하는 프로그램을 작성해보자. (if-else 문 사용)

입력	출력
2 3	큰 수는 3
21 21	큰 수는 21

목차

- 1) 조건문 개요
- 2) if 문
- 3) if-else 문
- 4) **다중 if 문**
- 5) switch 문

4) 다중 if 문

- 다중 if 문 or 중첩 if문

- if 문 안의 문장이 또 다른 if 문인 경우 (if-else 문 포함)

- 예 1) x가 0보다 같거나 크^면, 다음 두 문장 실행

- ✓ x의 값 출력

- ✓ x의 값이 짝수^{이면}, "양의 짝수 또는 0" 출력

```
if( x >= 0 ) {  
    printf("x의 값은 %d\n", x);  
    if( x%2 == 0 )  
        printf("양의 짝수 또는 0\n");  
}
```

⇒ 중괄호 생략하면 다른 의미가 됨 (왜?)

4) 다중 if 문

- 예 2) x가 0보다 같거나 크^면, 다음 두 문장 실행
 - ✓ x의 값 출력
 - ✓ x의 값이 짝수^{이면}, "양의 짝수 또는 0" 출력하고,
그렇지 않으면 "양의 홀수" 출력

```
if( x >= 0 ) {  
    printf("x의 값은 %d\n", x);  
    if( x%2 == 0 )  
        printf("양의 짝수 또는 0\n");  
    else  
        printf("양의 홀수\n");  
}
```

4) 다중 if 문

- 예 3) x가 0보다 같거나 크~~면~~, 다음 문장 실행
 - ✗ ~~x의 값 출력 (삭제)~~
 - ✓ x의 값이 짝수이면, "양의 짝수 또는 0" 출력하고, 그렇지 않으면 "양의 홀수" 출력


```
if( x >= 0 ) {  
  
    if( x%2 == 0 )  
        printf("양의 짝수 또는 0\n");  
    else  
        printf("양의 홀수\n");  
} ⇨ 중괄호 생략 가능
```

» 안쪽의 if-else 는 하나의 문장으로 취급됨

4) 다중 if 문


- else는 위쪽에서 짝이 없는 가장 가까운 if와 짝이 된다.

```
if( x >= 0 )  
    if( x%2 == 0 )  
        printf("양의 짝수 또는 0\n");  
    else  
        printf("양의 홀수\n");
```



- else를 멀리 있는 if와 짝을 지으려면?
 - 중괄호 { }를 사용하여 if문의 적용 범위를 강제로 바꿔주면 됨

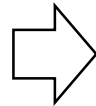
```
if(x >= 0) {  
    if(x%2 == 0)  
        printf("양의 짝수 또는 0\n");  
}  
else printf("음수\n");
```



4) 다중 if 문

- 정수를 4 개로 분류하는 다음 코드를 else의 짝에 맞추어 적절히 들여쓰기 하면?

```
if(x >= 0)
if(x%2 == 0)
printf("양의 짝수\n");
else
printf("양의 홀수\n");
else
if(x%2 == 0)
printf("음의 짝수\n");
else
printf("음의 홀수\n");
```



```
if(x >= 0)
    if(x%2 == 0)
        printf("양의 짝수\n");
    else
        printf("양의 홀수\n");
else
    if(x%2 == 0)
        printf("음의 짝수\n");
    else
        printf("음의 홀수\n");
```


4) 다중 if 문

- 초등 1, 2학년을 위한 퀴즈 프로그램을 작성하시오.
 - 다음 페이지에 있는 순서도를 참고하시오.

몇 학년 입니까(1, 2)? 1

세종대왕이 만드신 배는 거북선이다. 맞으면 1, 틀리면 0: 1
틀렸습니다

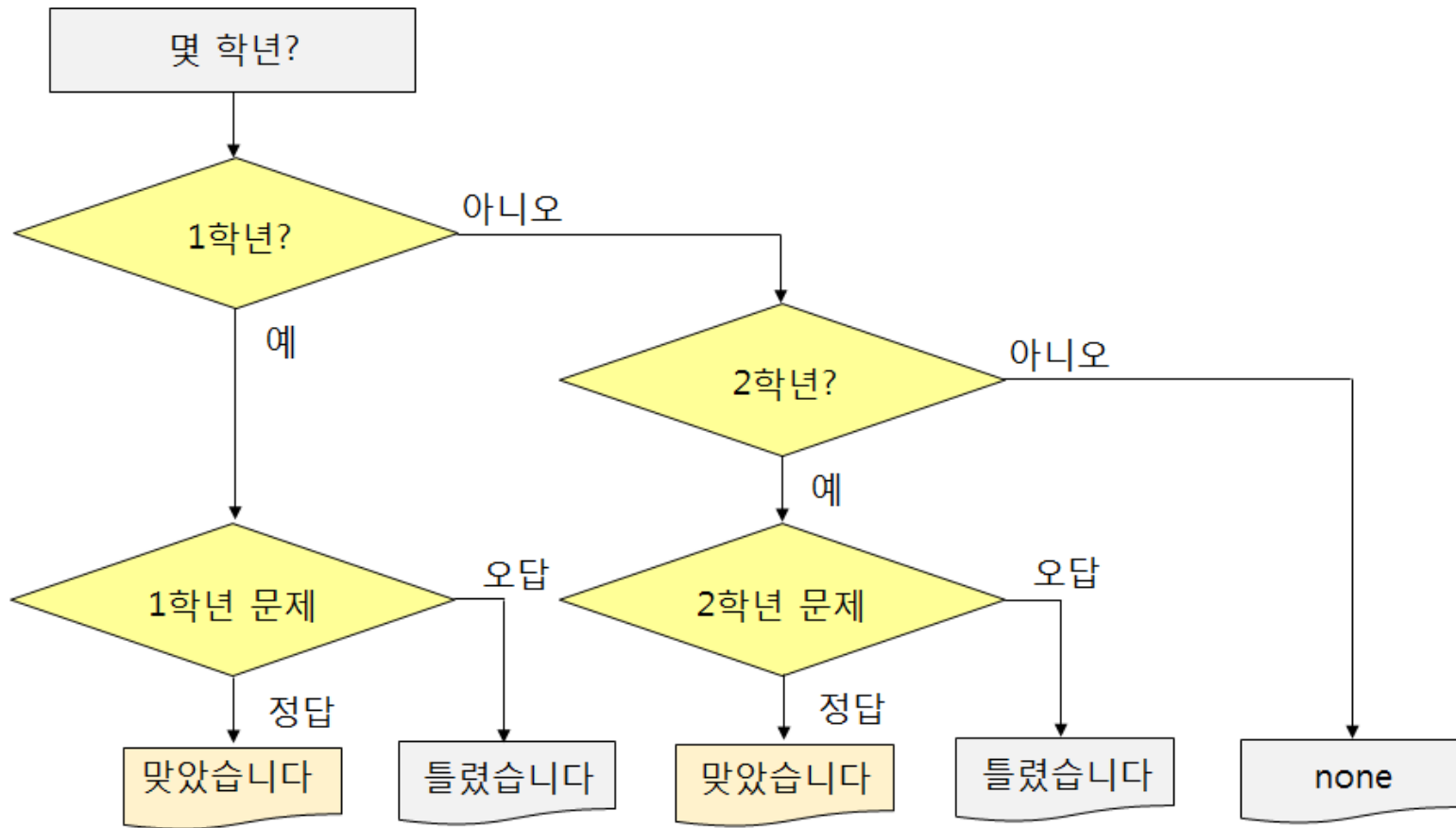
몇 학년 입니까(1, 2)? 2

35 * 70 = 123

틀렸습니다

4) 다중 if 문

- 초등 1, 2학년을 위한 퀴즈 프로그램



4) 다중 if 문

- 다중 선택을 위한 다중 if문

- 다음과 같이 점수에 따라 학점을 출력하는 프로그램을 다중 if문을 사용하여 작성할 수 있다. (다음 슬라이드 참조)

- ① 이번 학기 점수를 입력하라고 함(점수는 0~100점 사이)
- ② 만약 점수가 **90점 이상**이면 A학점을 출력함
- ③ 만약 점수가 **80점 이상~90점 미만**이면 B학점을 출력함
- ④ 만약 점수가 **70점 이상~80점 미만**이면 C학점을 출력함
- ⑤ 만약 점수가 **60점 이상~70점 미만**이면 D학점을 출력함
- ⑥ 만약 점수가 **60점 미만**이면 F학점을 출력함

4) 다중 if 문

- 다중 선택을 위한 다중 if문
 - 앞서 배운 대로, 각 if 와 else 짝을 맞추어 들여쓰기 한 버전

```
if(score>=90)
    printf("A\n");
else
    if(score>=80)
        printf("B\n");
    else
        if(score>=70)
            printf("C\n");
        else
            if(score>=60)
                printf("D\n");
            else
                printf("F\n");
```

4) 다중 if 문

- 다중 선택을 위한 다중 if문

- 하지만, 이런 다중 선택을 위한 다중 if문이 여러 개 겹치는 경우, 들여쓰기를 하지 않는 것이 이해하기 쉬움(구문은 동일)

```
if(score>=90)
    printf("A\n");
else if(score>=80)
    printf("B\n");
else if(score>=70)
    printf("C\n");
else if(score>=60)
    printf("D\n");
else
    printf("F\n");
```

4) 다중 if 문

- 다중 선택을 위한 다중 if문의 일반적인 구문

```
if( 조건식 A ) {  
    문장1;  ⇨ 조건식 A가 참일 때 실행  
}  
else if( 조건식 B ) {  
    문장2;  ⇨ 조건식 A는 거짓이고, 조건식 B는 참일 때 실행  
}  
else if( 조건식 C ) {  
    문장3;  ⇨ 조건식 A, B는 모두 거짓이고, 조건식 C는 참일 때 실행  
}  
...  
else {  
    문장;    ⇨ 조건식 A, B, C, ... 모두 거짓일 때 실행  
} ⇨ 각 블록 안의 문장이 하나인 경우, 해당 블록의 중괄호 생략 가능
```

4) 다중 if 문

- [예제 5.5]

- 다중 if문을 이용하여 다음 프로그램을 작성하시오.

- ① 사용자로부터 문자 하나를 입력 받는다.
- ② 문자가 대문자일 경우 “대문자”를 출력한다.
- ③ 문자가 소문자일 경우 “소문자”를 출력한다.
- ④ 문자가 숫자일 경우 “숫자”를 출력한다.
- ⑤ 그 밖의 경우 “기타”를 출력한다.

4) 다중 if 문

- 참고) 다중 if문의 들여쓰기는 논리적인 흐름에 맞추어 적절히 선택
 - 예) 정수를 4가지로 분류하는 프로그램
 - ✓ 어느 쪽이 논리적으로 이해하기 쉽고 명확한가?

```
if(x >= 0)
    if(x%2 == 0)
        printf("양의 짝수\n");
    else
        printf("양의 홀수\n");
else
    if(x%2 == 0)
        printf("음의 짝수\n");
    else
        printf("음의 홀수\n");
```

```
if(x >= 0)
    if(x%2 == 0)
        printf("양의 짝수\n");
    else
        printf("양의 홀수\n");
else if(x%2 == 0)
    printf("음의 짝수\n");
else
    printf("음의 홀수\n");
```


목차

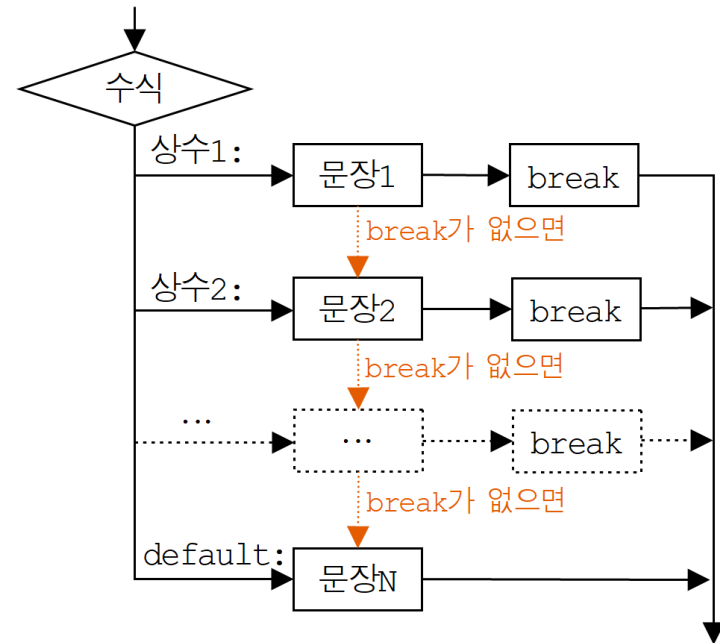
- 1) 조건문 개요
- 2) if 문
- 3) if-else 문
- 4) 다중 if 문
- 5) **switch 문**

5) switch 문

▪ switch 문

- 주로 다중 선택에 사용하는데, 다중 if문보다 간결
- 대신, 사용 영역이 if-else에 비해 제한적
- 구문

```
switch( 수식 ) {  
    case 상수1:  
        문장1 ;  
        break; ⇨ 필요시  
    case 상수2:  
        문장2 ;  
        break; ⇨ 필요시  
    ...  
    default: ⇨ 필요시  
        문장N;  
}
```



5) switch 문

- if~else 문을 switch 문으로 바꿔보자

```
if(x==1)
    printf("January\n");
else if(x==2)
    printf("February\n");
else if(x==3)
    printf("March\n");
else
    printf("None\n");
```



```
switch(x){
    case 1:
        printf("January\n");
        break;
    case 2:
        printf("February\n");
        break;
    case 3:
        printf("March\n");
        break;
    default:
        printf("None\n");
        break;
}
```

5) switch 문

▪ switch 문 동작 방식

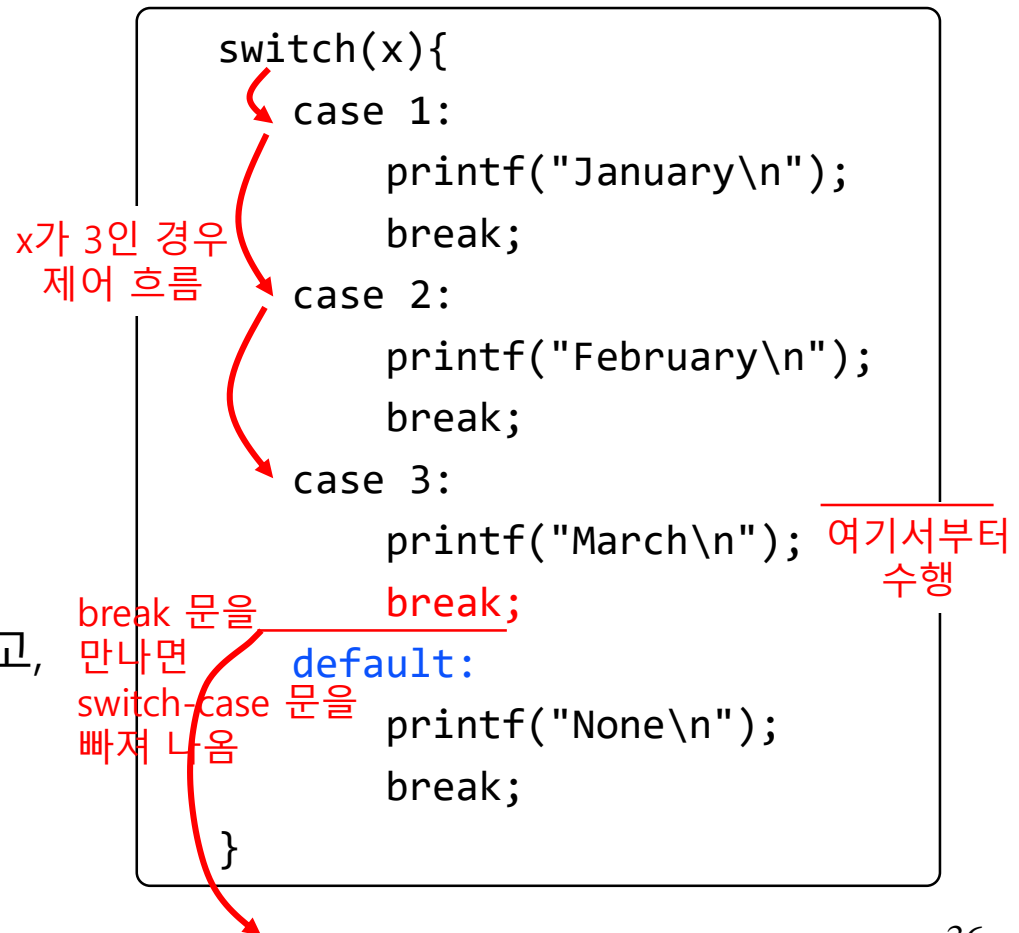
- switch문 **수식의 결과 값**과 **case의 상수 값을 차례로 비교**하여 일치하는 case의 문장들부터 차례로 수행

- ✓ 일치하는 case 이전의 문장들은 무시됨

- case 비교 시 **default:** 의 의미는?

- ✓ 그 외의 경우

- 해당 case도 없고, default: 도 없으면?
 - ✓ 아무 문장도 실행하지 않고, switch 문 끝으로 이동



5) switch 문

▪ break 문

- switch 문을 중간에 벗어나는 역할
- switch 문 수행도중 break문을 만나면 switch문 끝으로 이동
- break 문을 제거하면?

실행 결과

March
April
None

```
x = 3;
```

```
switch(x){  
    case 1:    printf("January\n");  
    case 2:    printf("February\n");  
    case 3:    printf("March\n");  
    case 4:    printf("April\n");  
    default:   printf("None\n");  
}
```

⇒ break가 없으면?

5) switch 문

- **[예제 5.6]** 학점 출력 문제를 switch 문을 사용하여 구현하시오.
 - Hint: $\text{score}/10=91/10=9$, 정수/정수=정수

```
if(score>=90)
    printf("A\n");
else if(score>=80)
    printf("B\n");
else if(score>=70)
    printf("C\n");
else if(score>=60)
    printf("D\n");
else
    printf("F\n");
```

5) switch 문

- 답안

- case 10과 case 9 사용 형태를 눈여겨 보자.

```
switch(score/10){  
    case 10: case 9: printf("A\n"); break;  
    case 8:  printf("B\n"); break;  
    case 7:  printf("C\n"); break;  
    case 6:  printf("D\n"); break;  
    default: printf("F\n");  
}
```

5) switch 문

- case 에 정수 1, 2, 3이 순서대로 나와야 하나? NO!!
 - 순서 상관없다 (하지만, 굳이..)
 - 문자도 가능

```
switch(ch){  
    case 'A':    printf("잘했습니다.\n"); break;  
    case 'C':    printf("괜찮습니다.\n"); break;  
    case 'B':    printf("좋습니다.\n"); break;  
    default:     printf("노력합시다.\n");  
}
```


5) switch 문

- switch 문 사용시 주의점

- case 에는 정수형 상수(문자 포함)만 가능

- ✓ case 1 : ⇒ (O) 정수형 상수
- ✓ case 'a' : ⇒ (O) 문자형 상수 (C언어에서 문자도 정수이다.)

- ✓ case 2.0 : (X) ⇒ (X) 부동소수 안됨.
- ✓ case 3+4 : (X) ⇒ (X) 수식 안됨.
- ✓ case a : (X) ⇒ (X) 변수 안됨. a는 변수, 'a'는 문자
- ✓ case "a" : (X) ⇒ (X) 문자열 안됨. "a"는 문자열(문자열은 10장에서)