

## 4. 흐름제어 - 조건문

Autumn 2019

# Today

1. 지난 시간 실습 복습
2. 블록과 복합문장
3. 흐름제어 – 조건문1 (if)
4. 흐름제어 – 조건문2 (switch)
5. 실습

# [복습] 데이터 유형

- 다양한 크기
  - **char**
  - **short**
  - **long**
  - **float**
  - **double**
  
- 숫자 유형
  - **signed / unsigned**
  
- 각종 연산자들
  - 수리연산, 증감연산, 대입연산,
  - 관계연산, 논리연산, 비트연산
  
- 연산 우선순위 순서

# Today

1. 지난 시간 실습 복습
2. 블록과 복합문장
3. 흐름제어 – 조건문1 (if)
4. 흐름제어 – 조건문2 (switch)
5. 실습

# 블록(blocks)과 복합문장

- 단순문장은 semicolon (; ) 으로 끝남
  - `z = foo (x+y);`
- 위 문장은 여러 문장으로 다음과 같이 기술함
  - `temp = x + y;`
  - `z = foo (temp);`
- 중괄호 { }
  - 복합문장 또는 블록을 한 단위로 묶어주는 기능을 함

# 블록(blocks)

- 블록은 단순문장으로 대체할 수 있음
- 하나의 단위로 컴파일됨
- 변수들은 중괄호 안에서 선언함
- 블록 끝은 **세미콜론을 쓰지 않음**

```
{  
    int temp = x+y;  
    z = foo(temp);  
}
```

# 블록 안에 블록(nested blocks)

- 블록 안에 블록을 위치할 수 있음

```
{  
    int temp = x+y;  
    z = foo(temp);  
    {  
        float temp2 = x*y;  
        z += bar(temp2);  
    }  
}
```

# Today

1. 지난 시간 실습 복습
2. 블록과 복합문장
3. 흐름제어 – 조건문1 (if)
4. 흐름제어 – 조건문2 (switch)
5. 실습



# 제어 조건들 (control conditions)

- 조건에 따라 선택하는 것을 표현하는 경우
- 조건 (condition)은 표현식임
  - 예를 들어,  
 $n < 3$  또는  $x < y \parallel z < y$
- 조건이 non-zero 이면  $\rightarrow$  조건은 '참'(true) 을 의미함
- ...

# 조건문 (Conditional statements)

- 조건문 종류
  - The `if` statement
  - The `switch` statement

# if 문 (The if statement)

```
if (x % 2 == 0)  
    y += x / 2;
```

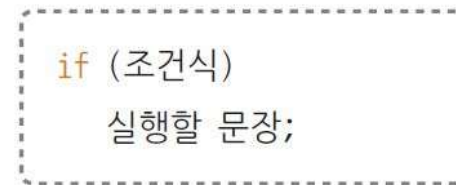
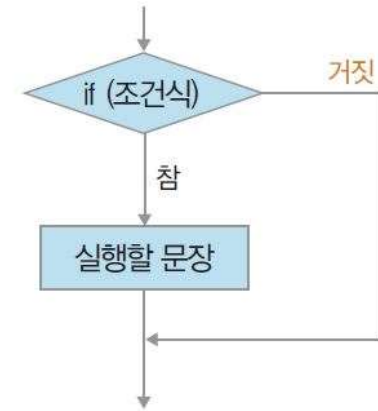


그림 5-1 if문의 형식과 순서도



- 조건을 평가함

`if (x % 2 == 0)`

- 만약, 조건이 '참'이면, 내부 블록을 수행함

`y += x / 2;`

- '참'이 아니면, 아무것도 수행하지 않음

- if문에서 두 문장 이상을 실행하고 싶다면, 여러 개의 구문을 하나로 만드는 블록({ })을 사용한다.

## [5-1] 기본 if문 사용 예 ①

```
01 #include <stdio.h>
02
03 int main()
04 {
05     int a = 99;
06
07     if (a < 100)
08         printf ("100보다 작군요..\n");
09 }
```

---a가 100보다 작으므로 참이다.

- 정수형 변수 a 에 99 라는 값을 할당
- if 문의 조건식  $a < 100$  의 '참'/'거짓' 여부를 판단함
- '참' 이므로 if 내부의 문장을 실행함

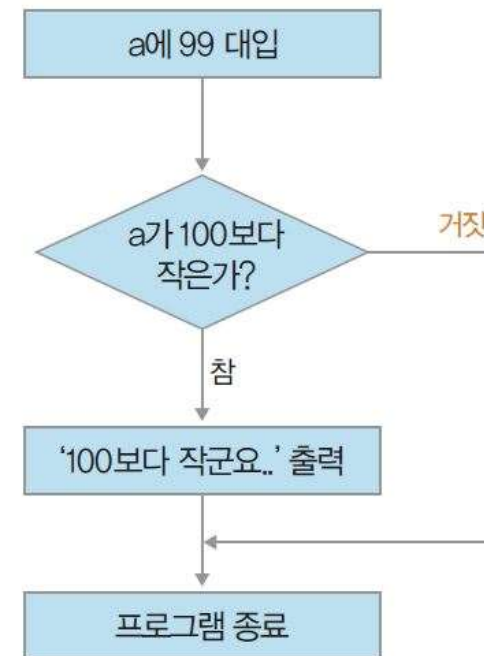


그림 5-2 [기본 5-1]의 실행 과정

## [기본 5-2] 기본 if문 사용 예 ②

```
01 #include <stdio.h>
02
03 int main()
04 {
05     int a = 200;
06
07     if (a < 100)
08         printf ("100보다 작군요...\n");
09         printf ("거짓이므로 이 문장은 안보이겠조?\n");
10
11     printf ("프로그램 끝!\n");
12 }
```

---7행이 참일 경우에 수행할 것으로 예상된다.

---7행이 거짓이면 8~9행을 수행하지 않고 11행만 수행할 것으로 예상된다.

```
01 #include <stdio.h>
02
03 int main()
04 {
05     int a = 200;
06
07     if (a < 100)
08     {
09         printf ("100보다 작군요...\n");
10         printf ("거짓이므로 앞의 문장은 안보이겠조?\n");
11     }
12
13     printf ("프로그램 끝!\n");
14 }
```

---7행이 참이면 블록으로 감싼 부분이 모두 수행된다.

# else 예약어 (The else keyword)

```
if (x % 2 == 0)
    y += x / 2;
else
    y += (x+1) / 2;
```

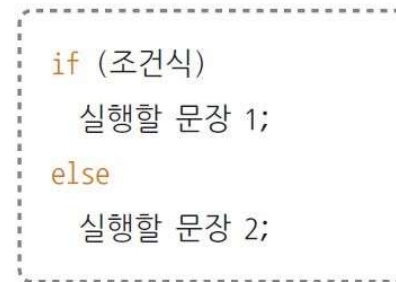
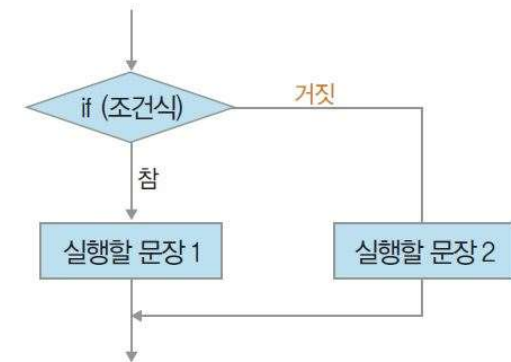


그림 5-3 if~else문의 형식과 순서도



- else 문은 선택사항임
- 만약, if 조건이 '거짓'이면, else 블록을 실행함  
 $y += (x+1) / 2;$

## [5-4] if~else문 사용 예

```
01 #include <stdio.h>
02
03 int main()
04 {
05     int a = 200;
06
07     if (a < 100)
08         printf ("100보다 작군요..\n");
09     else
10         printf ("100보다 크군요..\n");
11 }
```

---7행이 참이면(a가 100보다 작으면) 실행한다.

---7행이 거짓이면(a가 100보다 크거나 같으면) 실행한다.

- 정수형 변수 a 는 200이란 값을 할당함
- if 문의 조건은 '거짓'이므로 if의 블록을 수행하지 않고, else 의 블록을 수행함

## [5-5] 블록을 활용한 if~else문 사용 예 ①

```
01 #include <stdio.h>
02
03 int main()
04 {
05     int a = 200;
06
07     if (a < 100)
08     {
09         printf ("100보다 작군요..\n");
10         printf ("참이면 이 문장도 보이겠죠?\n");
11     }
12     else
13     {
14         printf ("100보다 크군요..\n");
15         printf ("거짓이면 이 문장도 보이겠죠?\n");
16     }
17
18     printf ("프로그램 끝! \n");
19 }
```

---7행이 참이면(a가 100보다 작으면) 실행한다.

---7행이 거짓이면(a가 100보다 크거나 같으면) 실행한다.



## [5-6] 블록을 활용한 if~else문 사용 예 ②

```
01 #include <stdio.h>
02
03 int main()
04 {
05     int a;
06
07     printf("정수를 입력하세요 : ");
08     scanf_s("%d", &a);      ---정수를 입력받는다.
09
10     if ( ____①____ == 0)    ---입력된 값을 2로 나눈 나머지값이 0일 경우
11     {                      ---짝수이면(2로 나눈 나머지값이 0이면) 실행한다.
12         printf ("짝수를 입력했군요..\n");
13     }
14     else
15     {                      ---홀수이면(2로 나눈 나머지값이 0이 아니면) 실행한다.
16         printf ("홀수를 입력했군요..\n");
17     }
18 }
```

## else if 예약어 (**The else if keyword**)

```
if (x % 2 == 0)
    y += x / 2;
else if (x % 4 == 1)
    y += 2 * ( (x+3) / 4 );
else
    y += (x+1) / 2;
```

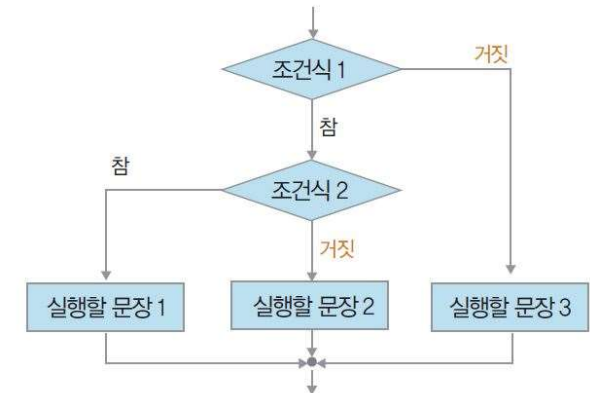
- 다른 제어를 추가적으로 처리하는 기능
- if 조건들 중 '참'인 것을 하나라도 만나면, 내부 블록을 수행함
- 만약, 여러 조건들이 '참'이면, 가장 첫번째 구문이 실행됨

# 중첩 if 문 (Nesting if statements)

```
if (x % 4 == 0)
    if (x % 2 == 0)
        y = 2;
else
    y = 1;
```

```
if (조건식 1){
    if (조건식 2)
        실행할 문장 1;
    else
        실행할 문장 2;
}
else
    실행할 문장 3;
```

그림 5-5 중첩 if문의 형식과 순서도



- if문을 한 번 실행하고 그 결과에 다시 다른 if문을 실행하는 것
- 어떤 if 문이 else 예약어와 짝을 이룬 것일까요?
- else 와 짝을 이루는 if 문은 **중괄호{ }를 써야함**

```
if (x % 4 == 0) {
    if (x % 2 == 0)
        y = 2;
} else
    y = 1;
```

가능하면 블록으로 조건식을 명확히 할 것!!!

## [5-7] 중첩 if문 사용 예 ①

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

```
02
```

```
03 int main()
```

```
04 {
```

```
05     int a = 75;
```

```
06
```

```
07     if (a > 50)
```

```
08     {
```

```
09         if (a < 100)
```

```
10         {
```

```
11             printf ("50보다 크고 100보다 작군요..\n");
```

```
12         }
```

```
13     else
```

```
14     {
```

```
15         printf ("와~~ 100보다 크군요..\n");
```

```
16     }
```

```
17 }
```

```
18 else
```

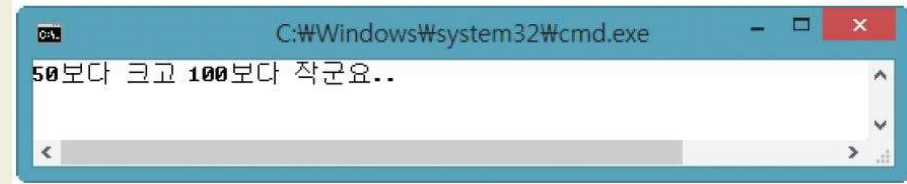
```
19 {
```

```
20     printf ("에게~ 50보다 작군요..\n");
```

```
21 }
```

```
22 }
```

실행 결과 ▼



----7행이 참이면(a가 50보다 크면) 실행한다.

-----7행이 참이고(a가 50보다 크고)  
9행이 참이면(a가 100보다 작으면) 실행한다.

-----7행이 참이고(a가 50보다 크고)  
9행이 거짓이면(a가 100보다 크거나 같으면)  
실행한다.

----7행이 거짓이면(a가 50보다 작거나 같으면) 실행한다.

## [5-8] 중첩 if문 사용 예 ②

```
01 #include <stdio.h>
02
03 int main()
04 {
05     int a;
06
07     printf("점수를 입력하세요 : ");
08     scanf("%d", &a);
09
10     if ( a >= 90 )
11         printf ("A");
12     else
13         if ( a >= 80 )
14             printf ("B");
15         else
16             if (a >= 70)
17                 printf ("C");
18             else
19                 if (a >= 60)
20                     printf ("D");
21             else
22                 printf ("F");
23
24     printf (" 학점 입니다. \n");
25 }
```

```
#include <stdio.h>

int main()
{
    int a;

    printf("점수를 입력하세요 : ");
    scanf("%d", &a);

    if (a >= 90)
        printf("A");
    else if (a >= 80)
        printf("B");
    else if (a >= 70)
        printf("C");
    else if (a >= 60)
        printf("D");
    else
        printf("F");
}
```

## [5-8] 중첩 if문 사용 예 ②

### ■ 10~23행

- a가 90점 이상일 때와 미만일 때를 먼저 구분
- 90점 미만이면 또 다시 조건식 사용

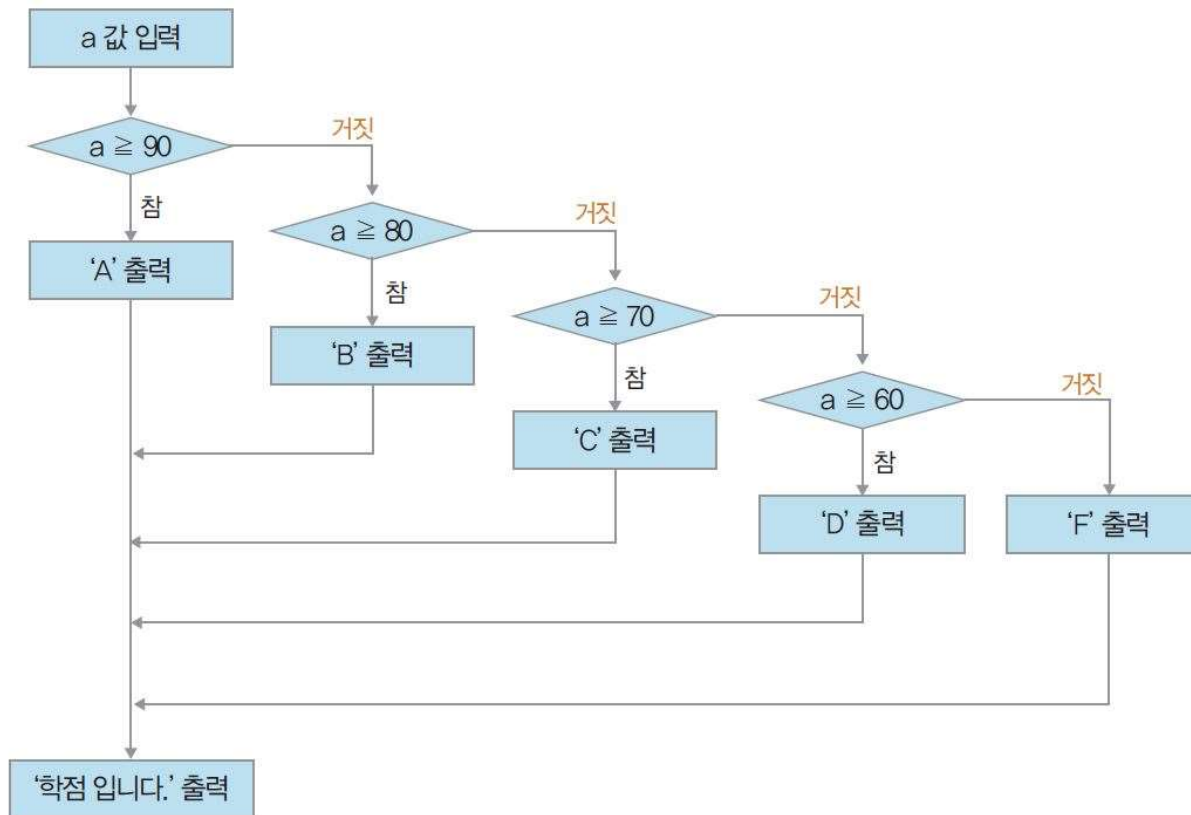


그림 5-6 [응용 5-8]의 실행 과정

# Today

1. 지난 시간 실습 복습
2. 블록과 복합문장
3. 흐름제어 – 조건문1 (if)
4. 흐름제어 – 조건문2 (switch)
5. 실습

# switch 문 (The switch statement)

- 여러 조건들을 처리하는 구문
- 정수 (또는 char 문자)를 변수의 값으로 입력받음
- 변수가 가질 수 있는 경우의 수(case)를 처리함

```
switch (ch) {  
    case 'Y': /* ch == 'Y' */  
        /* do something */  
        break;  
    case 'N': /* ch == 'N' */  
        /* do something else */  
        break;  
    default: /* otherwise */  
        /* do a third thing */  
        break;  
}
```

```
switch(정숫값){  
    case 정숫값 1:  
        실행할 문장 1;  
        break;  
    case 정숫값 2:  
        실행할 문장 2;  
        break;  
    default:  
        실행할 문장 3;  
        break;  
}
```

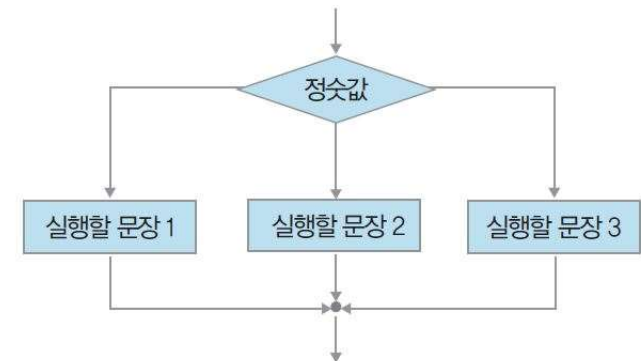


그림 5-7 switch~case문의 형식과 순서도



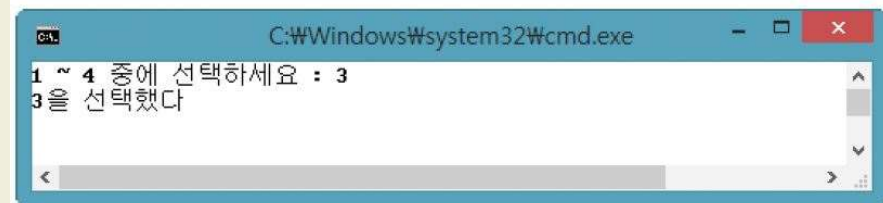
## [5-9] switch~case문 사용 예 ①

```
01 #include <stdio.h>
02
03 int main()
04 {
05     int a;
06
07     printf("1 ~ 4 중에 선택하세요 : ");
08     scanf("%d", &a);
09
10     switch (a)
11     {
12         case 1:
13             printf ("1을 선택했다\n");
14             break;
15         case 2:
16             printf ("2를 선택했다\n");
17             break;
18         case 3:
19             printf ("3을 선택했다\n");
20             break;
21         case 4:
22             printf ("4를 선택했다\n");
23             break;
24         default:
25             printf ("이상한걸 선택했다.\n");
26     }
27 }
```

---입력한 a 값에 따라서 분기한다.

---a가 1이면 13행을 수행하고  
14행에서 switch 블록을 빠져나간다

실행 결과 ▼



---a가 1, 2, 3, 4 중 아무것도 해당되지 않을 때 수행한다.

# switch ~ case 문

- 변수 **a** 에 입력값으로 **3**이 들어오는 경우,

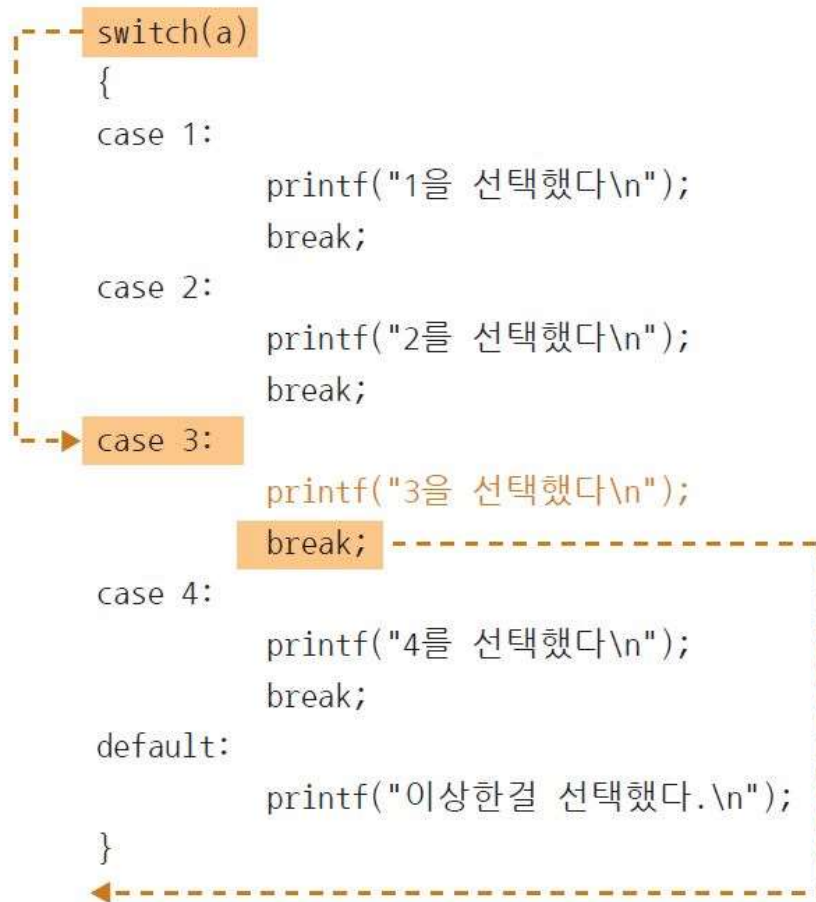


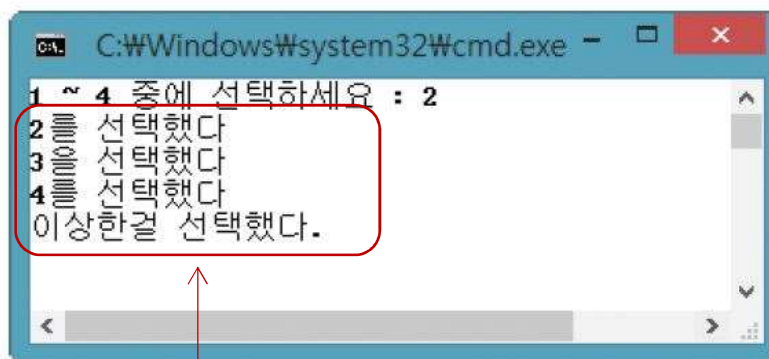
그림 5-8 a가 3일 때의 switch~case문 흐름도

# switch 에서 break 역할

- switch~case문을 빠져나가는 역할
- 실행문의 마지막에 반드시 써줘야 함
- [5-9]에서 **break**문을 빼고 실행한 결과

```
10 switch (a)
11 {
12 case 1:
13     printf ("1을 선택했다\n");
14 case 2:
15     printf ("2를 선택했다\n");
16 case 3:
17     printf ("3을 선택했다\n");
18 case 4:
19     printf ("4를 선택했다\n");
20 default:
21     printf ("이상한걸 선택했다.\n");
22 }
```

실행 결과 ▼



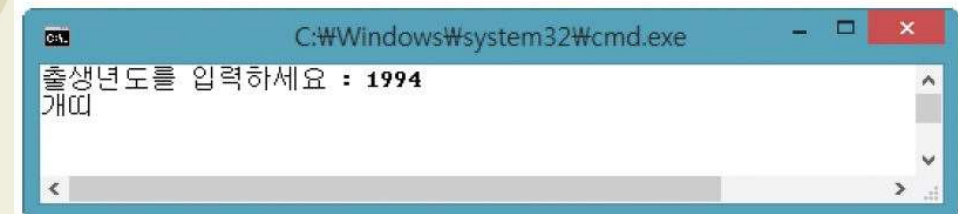
Break문이 없어 모든 실행문 출력

## [5-10] switch~case문 사용 예 ②

```
01 #include <stdio.h>
02
03 int main()
04 {
05     int year;
06
07     printf("출생년도를 입력하세요 : ");
08     scanf("%d", &year);
09
10     switch ( ____①____ )
11     {
12         case 0 : printf ("원숭이띠\n"); break;
13         case 1 : printf ("닭띠\n"); break;
14         case 2 : printf ("개띠\n"); break;
15         case 3 : printf ("돼지띠\n"); break;
16         case 4 : printf ("쥐띠\n"); break;
17         case 5 : printf ("소띠\n"); break;
18         case 6 : printf ("호랑이띠\n"); break;
19         case 7 : printf ("토끼띠\n"); break;
20         case 8 : printf ("용띠\n"); break;
21         case 9 : printf ("뱀띠\n"); break;
22         case 10 : printf ("말띠\n"); break;
23         case 11 : printf ("양띠\n"); break;
24     }
25 }
```

---입력한 연도를 12로 나눈 나머지값에 따라서  
결과를 출력한다.

실행 결과 ▼



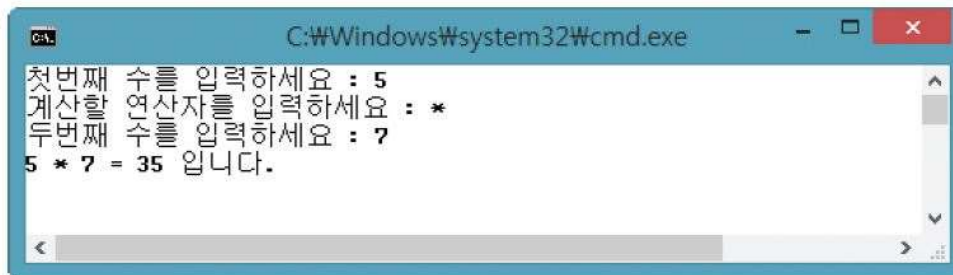
# Today

1. 지난 시간 실습 복습
2. 연산자1
3. 연산자2
4. 실습

# [실습1] 단순 if문을 활용한 간단한 계산기

**예제 설명** 단순 if문을 활용하여 두 수의 +, -, \*, /, % 연산을 수행하는 프로그램이다.

**실행 결과**



```
C:\Windows\system32\cmd.exe
첫번째 수를 입력하세요 : 5
계산할 연산자를 입력하세요 : *
두번째 수를 입력하세요 : 7
5 * 7 = 35 입니다.
```

## [실습2] 중복 if문을 활용한 간단한 계산기

**예제 설명** 중복 if문을 활용하여 두 수의 +, -, \*, /, % 연산을 수행하는 프로그램이다.

**실행 결과**



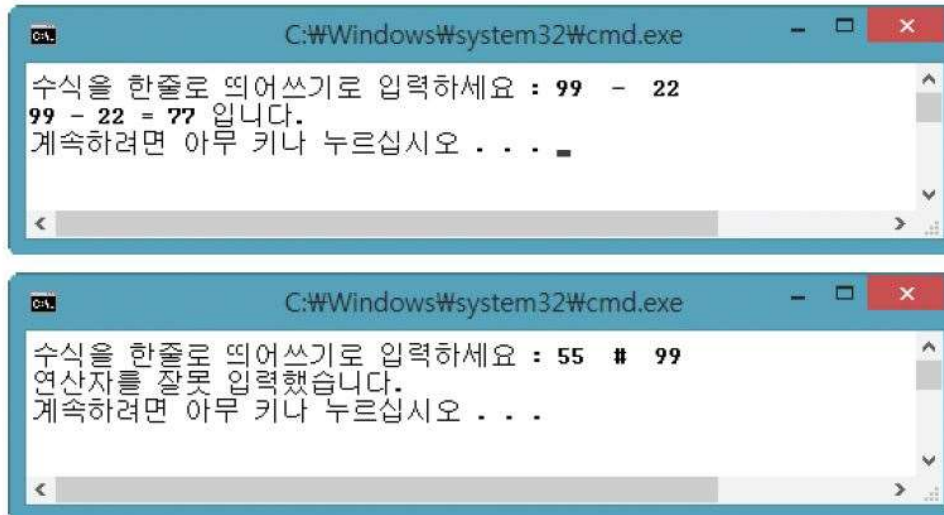
```
C:\Windows\system32\cmd.exe
첫번째 수를 입력하세요 : 12
계산할 연산자를 입력하세요 : /
두번째 수를 입력하세요 : 5
12 / 5 = 2.400000 입니다.

C:\Windows\system32\cmd.exe
첫번째 수를 입력하세요 : 88
계산할 연산자를 입력하세요 : &
두번째 수를 입력하세요 : ??
연산자를 잘못 입력했습니다.
```

## [실습3] switch~case문을 활용한 간단한 계산기

**예제 설명** 수식을 띄어쓰기로 한 줄에 입력받고 switch~case문을 활용하여 두 수의 +, -, \*, /, % 연산을 수행하는 프로그램이다.

**실행 결과**



```
C:\Windows\system32\cmd.exe
수식을 한줄로 띄어쓰기로 입력하세요 : 99 - 22
99 - 22 = 77 입니다.
계속하려면 아무 키나 누르십시오 . . .

C:\Windows\system32\cmd.exe
수식을 한줄로 띄어쓰기로 입력하세요 : 55 # 99
연산자를 잘못 입력했습니다.
계속하려면 아무 키나 누르십시오 . . .
```



# Summary

## ■ 조건문 if

- if 문은 <조건식>이 '참'일 때와 '거짓'을 판별하고, 해당하는 곳에서 블록을 수행함
- 중첩 if 문 또는 else if 문을 이용해서 추가 조건식을 처리할 수 있음
- 구조

```
if(조건식)
    { 참일 때 실행할 문장들 }
else
    { 거짓일 때 실행할 문장들 }
```

## ■ 조건문 switch

- 다양한 경우의 수가 있는 경우, 중첩if 문보다 깔끔하게 표현할 수 있다
- 구조

```
switch(정숫값){
    case 정숫값 1 :
        실행할 문장 1;
        break;
    case 정숫값 2 :
        실행할 문장 2;
        break;
    default :
        실행할 문장 3;
        break;
}
```