

4. 제어문

🔄 제어문이란?

프로그램의 흐름을 제어하여 원하는 문장만을 실행하거나 특정한 블록을 반복하여 실행 할 수 있도록 해주는 문법, 프로그램의 특정 부분으로 이동하거나 함수를 강제로 종료시키는 등의 일을 할 수 있다.

🔄 조건문

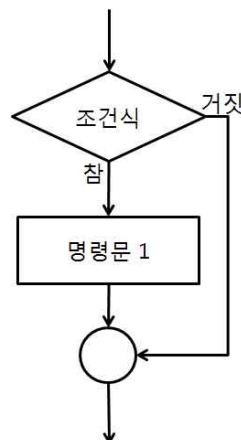
조건에 따라 실행할 명령이 결정되도록 해주는 문장이다. C언어에는 if문과 switch~ case라는 두 개의 조건문이 있으며 조건이 참(0이 아닌 값) 인지 거짓(0) 인지에 따라 원하는 문장만 실행 가능하도록 해준다. 조건에는 변수, 상수, 관계식, 함수 등을 모두 사용할 수 있다.

1. if 문

설명 : if 문은 특정 조건의 만족 여부에 따라 실행해야할 다음 명령문을 결정할 때 사용한다.

if (조건식)

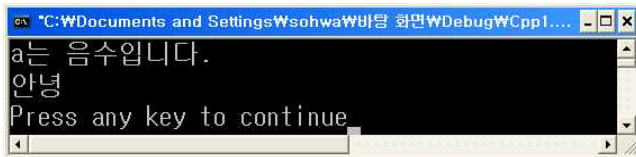
조건 만족 시 실행 할 문장;



의 형태로 사용하며 조건식이 참이라면 아래의 문장이 실행되고 조건이 거짓일 경우에는 문장이 실행되지 않는다. 조건을 만족 했을 때 실행하고자 하는 문장이 여러 개라면 { }로 묶어서 표시해야 한다. 필요시 else if와 else 문장을 추가하여 다양한 형태로 변경해 사용할 수 있다.

예제 4-1 : if 문의 사용방법 - 1

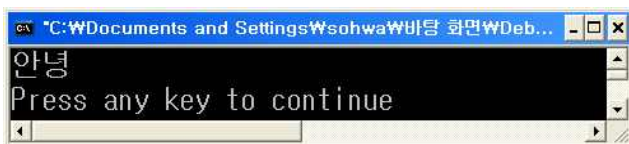
```
#include <stdio.h>
void main ( ) {
    int a = -2;
    if ( a < 0 )
        printf("a는 음수입니다.\n");
    printf("안녕\n");
}
```



```
C:\Documents and Settings\Wsohwa\바탕 화면\WDebug\WCpl1...
a는 음수입니다.
안녕
Press any key to continue
```

예제 4-2 : if 문의 사용방법 - 2

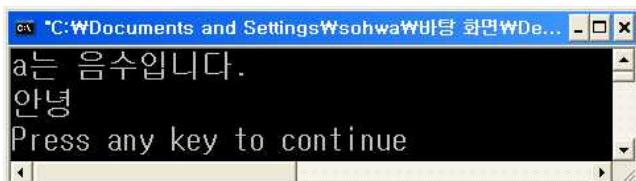
```
#include <stdio.h>
void main ( ) {
    int a = 2;
    if ( a < 0 )
        printf("a는 음수입니다.\n");
    printf("안녕\n");
}
```



```
C:\Documents and Settings\Wsohwa\바탕 화면\WDeb...
안녕
Press any key to continue
```

예제 4-3 : 여러 개의 문장을 if에 포함시키기 위해 { }를 사용한 예제

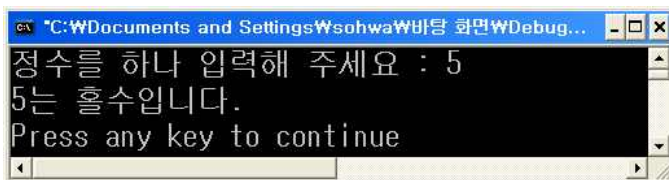
```
#include <stdio.h>
void main ( ) {
    int a = -12;
    if ( a < 0 ) {
        printf("a는 음수입니다.\n");
        printf("안녕\n");
    }
    if ( a > 0 ) {
        printf("a는 양수입니다.\n");
        printf("안녕\n");
    }
}
```



```
C:\Documents and Settings\Wsohwa\바탕 화면\WDe...
a는 음수입니다.
안녕
Press any key to continue
```

예제 4-4 : if문을 이용해 사용자가 입력한 수가 짝수인지 아닌지 판단하는 예제

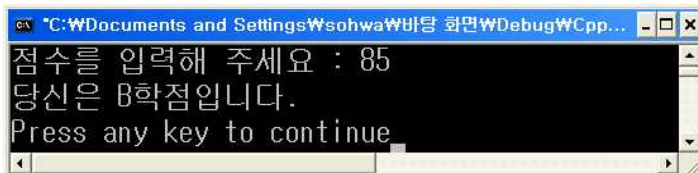
```
#include <stdio.h>
void main ( ) {
    int a;
    printf("정수를 하나 입력해 주세요 : ");
    scanf("%d", &a);
    if ( a % 2 == 0 ) {
        printf("%d는 짝수입니다.\n", a);
    }
    if ( a % 2 == 1 ) {
        printf("%d는 홀수입니다.\n", a);
    }
}
```



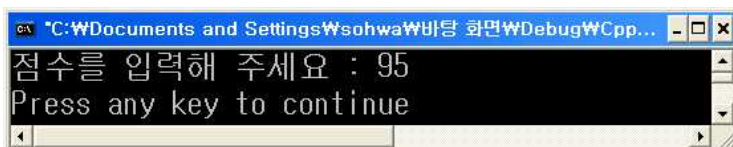
예제 4-5 : if 문의 조건을 숫자 범위로 주는 방법에 대한 예제

```
#include <stdio.h>
void main ( ) {
    int a;
    printf("점수를 입력해 주세요 : ");
    scanf("%d", &a);
    if ( a < 90 && a >= 80 ) {
        printf("당신은 B학점입니다.\n");
    }
}
```

입력 한 점수가 80 ~ 89점 사이일 경우의 실행결과



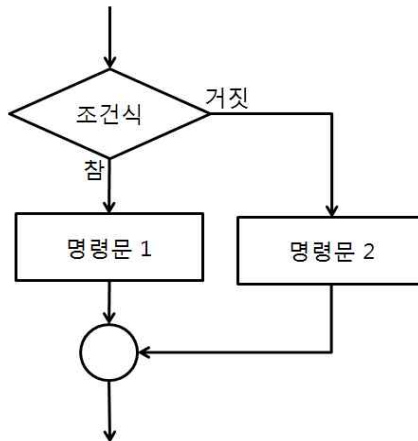
입력 한 점수가 80 ~ 89점 사이가 아닐 경우의 실행결과



2. if~ else~ 문

설명 : if~ else~ 문은 두 개의 명령문 중 조건이 만족한다면 명령문 1을 실행하고 그렇지 않으면 명령문 2를 실행하고자 할 경우에 사용한다. 사용방법은 아래와 같다.

```
if (조건식) {  
    명령문 1;  
}  
else {  
    명령문 2;  
}
```

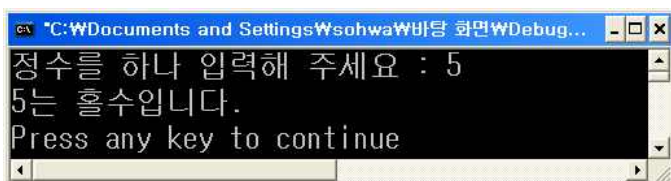


위와 같은 형태로 사용하며 조건식이 참이라면 명령문 1이 실행되고 조건이 거짓일 경우에는 명령문 2가 실행된다. if 와 else 모두 조건을 만족 했을 때 실행하고자 하는 문장이 여러 개라면 { }로 묶어서 표시해야 한다. else 따로 사용할 수 없으며 조건을 입력하지도 않는다. else는 조건이 참이 아닌 모든 경우를 말한다.

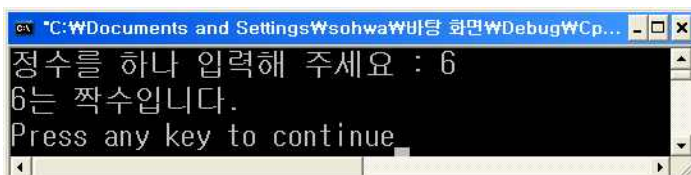
예제 4-6 : if문 else문을 이용해 사용자가 입력한 수가 짝수인지 아닌지 판단하는 예제

```
#include <stdio.h>  
void main ( ) {  
    int a;  
    printf("정수를 하나 입력해 주세요 : ");  
    scanf("%d", &a);  
    if ( a % 2 == 0 ) {  
        printf("%d는 짝수입니다.\n", a);  
    }  
    else {  
        printf("%d는 홀수입니다.\n", a);  
    }  
}
```

홀수를 입력했을 때의 실행 모양



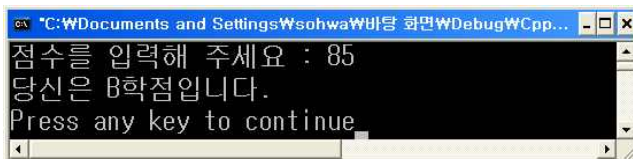
짝수를 입력했을 때의 실행 모양



예제 4-7 : if~ else~ 문을 이용해 조건을 숫자 범위로 주는 방법에 대한 예제

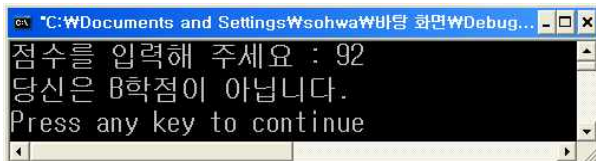
```
#include <stdio.h>
void main ( ) {
    int a;
    printf("점수를 입력해 주세요 : ");
    scanf("%d", &a);
    if ( a < 90 && a >= 80 ) {
        printf("당신은 B학점입니다.\n");
    }
    else {
        printf("당신은 B학점이 아닙니다.\n");
    }
}
```

입력 한 점수가 80 ~ 89점 사이일 경우의 실행결과



```
C:\Documents and Settings\Wsohwa\바탕 화면\WDebug\Cpp...
점수를 입력해 주세요 : 85
당신은 B학점입니다.
Press any key to continue
```

입력 한 점수가 80 ~ 89점 사이가 아닐 경우의 실행결과

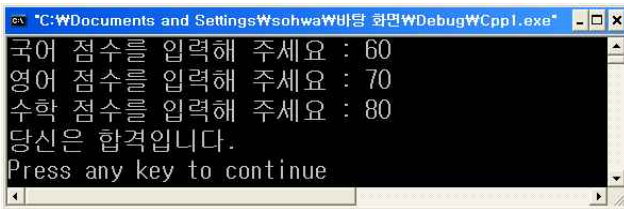


```
C:\Documents and Settings\Wsohwa\바탕 화면\WDebug...
점수를 입력해 주세요 : 92
당신은 B학점이 아닙니다.
Press any key to continue
```

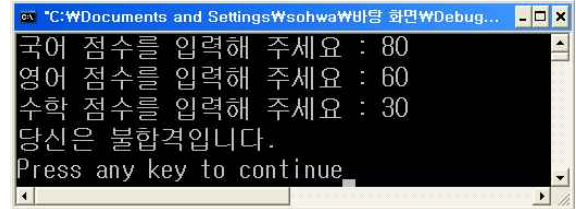
예제 4-8 : 점수를 입력받아 합격여부를 판단하는 예제. 국어, 영어, 수학의 3과목을 입력받아야 하며 한 과목이라도 40점 이하이면 불합격이고 세 과목 평균이 60이상이어야 한다.

```
#include <stdio.h>
void main ( ) {
    int kor, eng, mat;
    double ave; // 평균을 저장하기 위한 변수
    printf("국어 점수를 입력해 주세요 : ");
    scanf("%d", &kor);
    printf("영어 점수를 입력해 주세요 : ");
    scanf("%d", &eng);
    printf("수학 점수를 입력해 주세요 : ");
    scanf("%d", &mat);
    ave = (kor + eng + mat) / 3.0; // 평균을 구해 ave에 저장
    if ( ave >= 60 && kor >= 40 && eng >= 40 && mat >= 40 ) {
        printf("당신은 합격입니다.\n");
    }
    else {
        printf("당신은 불합격입니다.\n");
    }
}
```

점수가 합격일 경우의 실행 결과



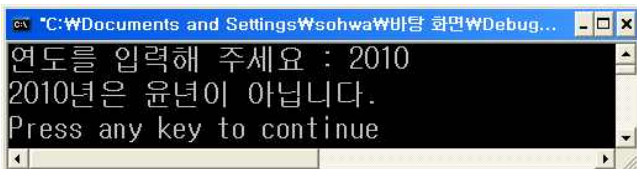
점수가 불합격일 경우의 실행결과



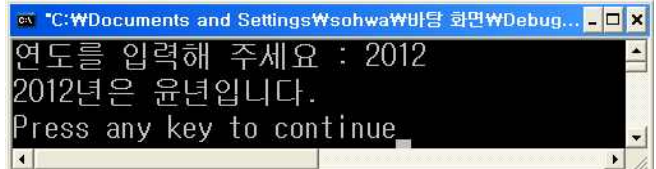
예제 4-9 : 연도를 입력받아 윤년인지 평년인지 판단하는 프로그램. 1년은 365.2422일 이지만 달력은 365일임으로 매년 0.2422일이 남는다. 이런 0.2422일에 400을 곱하면 대략 97일이 된다. 따라서 400년에 윤년은 97번 돌아온다. 400년에 97일을 나누어 놓으려면 매 4년마다 한 번씩은 윤년이 돌아와야 하며 매 100년은 윤년에서 제외되어야 한다. 그렇다면 총 96일임으로 400년이 되는 해는 윤년이 되어야 한다.

```
#include <stdio.h>
void main ( ) {
    int year;
    printf("연도를 입력해 주세요 : ");
    scanf("%d", &year);
    if ( year%4==0 && year%100!=0 || year%400==0) {
        printf("%d년은 윤년입니다.\n", year);
    }
    else {
        printf("%d년은 윤년이 아닙니다.\n", year);
    }
}
```

윤년이 아닐 경우의 실행 결과



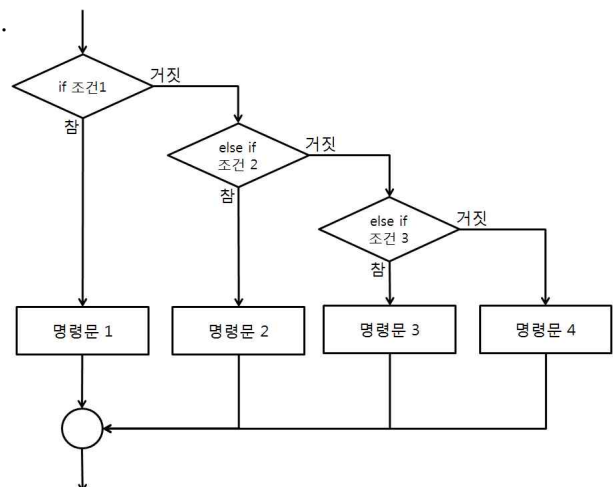
윤년일 경우의 실행 결과



3. 다중 if 문

설명 : if 문에서 여러 가지의 조건을 부여하고 싶을 때 사용한다. 추가할 조건은 else if문 뒤쪽에 기재하면 되며 사용방법은 아래와 같다.

```
if (조건식) {
    명령문 1;
}
else if (조건식 2) {
    명령문 2;
}
else if (조건식 3) {
    명령문 3;
}
else {
    명령문 4;
}
```



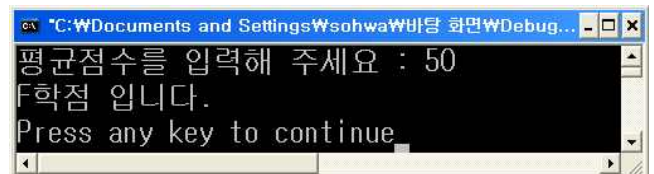
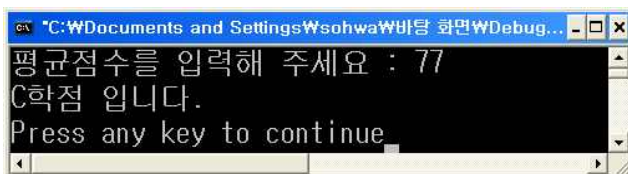
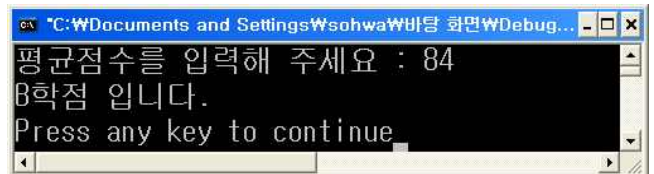
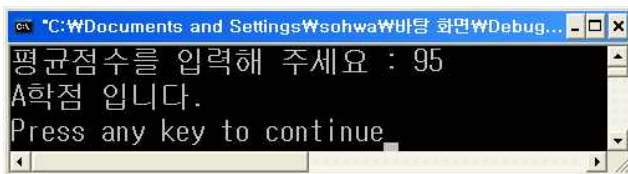
위와 같은 형태로 사용하며 조건식 1이 참이라면 명령문 1이 실행되고 종료한다. 하지만 조건식 1이 거짓일 경우에는 else if 뒤쪽의 조건문2가 실행된다. 조건식 2가 참이라면 명령문2를 실행하고 종료하고 조건식 2도 거짓일 때는 조건을 3이 실행된다. 조건식 3이 참이라면 명령문 3이 실행되고 종료한다. 조건식 3도 거짓이라면 else문이 실행된다. else if도 if와 마찬가지로 조건을 만족했을 때 실행하고자 하는 문장이 여러 개라면 { }로 묶어서 표시해야 한다. else if는 반드시 if문의 아래쪽에 사용되어야 하며 사용할 수 있는 else if의 개수에는 제한이 없다.

예제 4-10 : else if문을 사용해 입력받은 평균 점수에 해당하는 학점을 구해주는 프로그램

```
#include <stdio.h>

void main ( ) {
    double jumsu;
    printf("평균점수를 입력해 주세요 : ");
    scanf("%lf", &jumsu);
    if ( jumsu >= 90 ) {
        printf("A학점 입니다.\n");
    }
    else if ( jumsu >= 80 ) {
        printf("B학점 입니다.\n");
    }
    else if ( jumsu >= 70 ) {
        printf("C학점 입니다.\n");
    }
    else {
        printf("F학점 입니다.\n");
    }
}
```

입력한 점수별 실행 결과



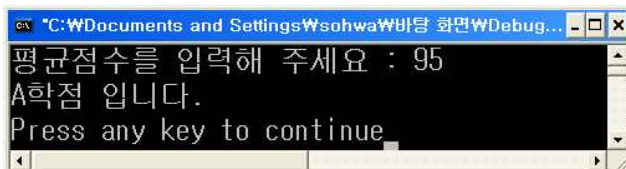
프로그램 설명 : else if문은 if의 조건이 틀렸을 때에만 실행된다. 따라서 위의 예제에 있는 if의 조건인 90점 이상을 만족한다면 A학점이 출력되고 프로그램이 종료된다. 하지만 90점 이상이 아니라면 else if의 조건식을 실행하여 80점 이상인지를 판단

한다. 80점 이상이라면 if가 거짓이었음으로 평균은 90보다 작고 80보다는 크거나 같다는 뜻이 된다.

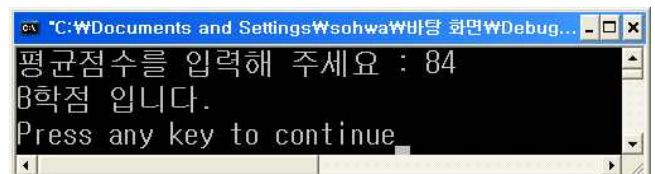
예제 4-11 : if문만을 사용해 위의 예제와 같은 결과를 보이는 프로그램

```
#include <stdio.h>
void main ( ) {
    double jumsu;
    printf("평균점수를 입력해 주세요 : ");
    scanf("%lf", &jumsu);
    if ( jumsu >= 90 ) {
        printf("A학점 입니다.\n");
    }
    if ( jumsu < 90 && jumsu >= 80 ) {
        printf("B학점 입니다.\n");
    }
    if ( jumsu < 80 && jumsu >= 70 ) {
        printf("C학점 입니다.\n");
    }
    if ( jumsu < 70 ) {
        printf("F학점 입니다.\n");
    }
}
```

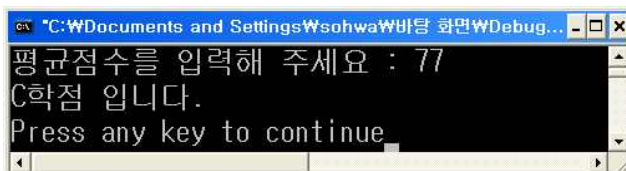
입력한 점수별 실행 결과



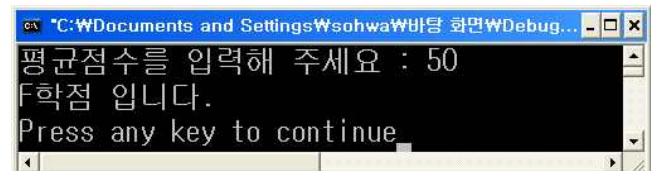
```
C:\WDocuments and Settings\Wsohwa\바탕 화면\WDebug...
평균점수를 입력해 주세요 : 95
A학점 입니다.
Press any key to continue
```



```
C:\WDocuments and Settings\Wsohwa\바탕 화면\WDebug...
평균점수를 입력해 주세요 : 84
B학점 입니다.
Press any key to continue
```



```
C:\WDocuments and Settings\Wsohwa\바탕 화면\WDebug...
평균점수를 입력해 주세요 : 77
C학점 입니다.
Press any key to continue
```

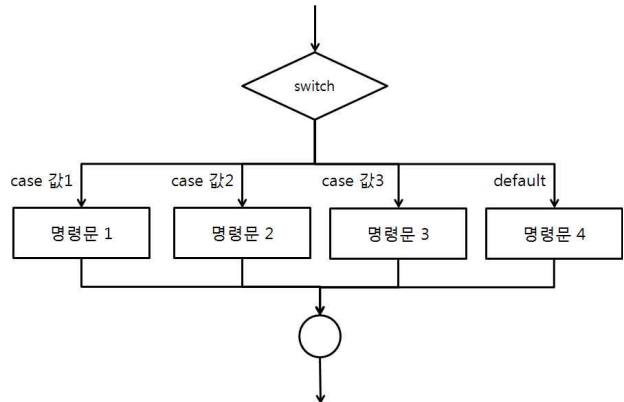


```
C:\WDocuments and Settings\Wsohwa\바탕 화면\WDebug...
평균점수를 입력해 주세요 : 50
F학점 입니다.
Press any key to continue
```


3. switch ~ case 문

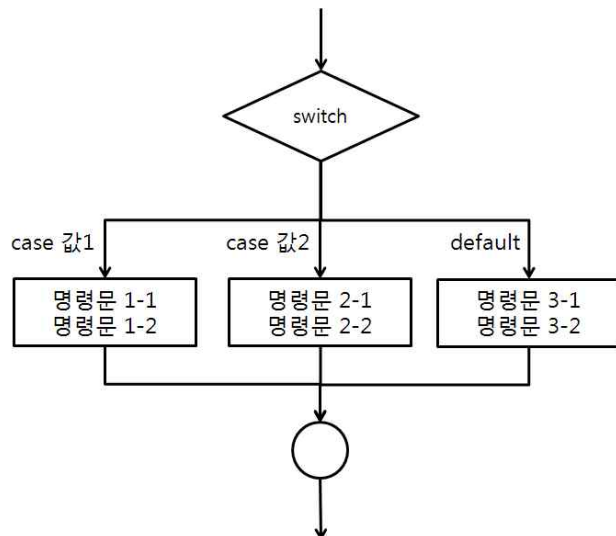
설명 : 다중 if문과 비슷하지만 보다 간결하게 여러 가지의 조건을 입력할 수 있다. 다중 if문은 매번 필요한 조건식을 입력해야 하지만 switch문은 조건에 해당하는 값만 기술해 제어문을 표현하며 사용되는 형태는 아래와 같다.

```
switch ( 상태 ) {
    case 값1 : 명령문 1; break;
    case 값2 : 명령문 2; break;
    case 값3 : 명령문 3; break;
    default  : 명령문 4; break;
}
```



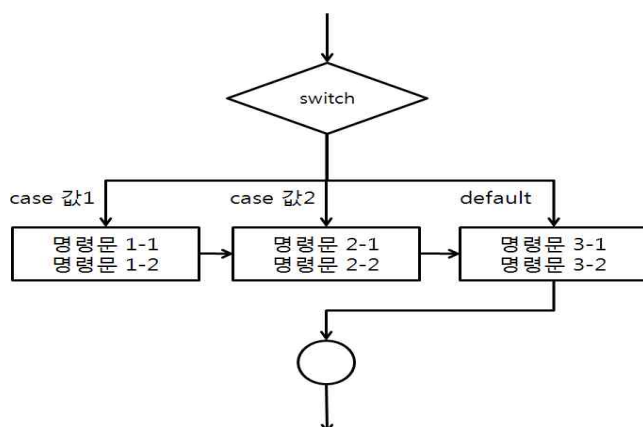
case를 만족했을 경우 실행하고 싶은 문장이 여러 개 라면 아래의 그림처럼 사용한다. (switch ~ case문은 조건 만족 시 실행할 문장이 여러 개라 할지라도 { }를 쓸 필요가 없다.)

```
switch ( 상태 ) {
    case 값1 :
        명령문 1-1;
        명령문 1-2;
        break;
    case 값2 :
        명령문 2-1;
        명령문 2-2;
        break;
    default :
        명령문 3-1;
        명령문 3-2;
        break;
}
```



switch ~ case문은 switch 뒤쪽의 () 안에 있는 상태의 결과와 case 뒤에 있는 값들 중 일치하는 것이 있는지를 찾아 일치하는 case가 있다면 해당 case 뒤쪽의 명령문이 실행된다. 명령문은 break 문을 만날 때 까지 연달아 실행되기 때문에 { }를 사용하지 않고도 여러 개의 명령문을 실행할 수가 있다. break 문을 쓰지 않으면 다음 case에 해당하는 문장도 모두 실행되게 된다. break가 없는 switch문은 아래의 그림과 같이 실행된다.

```
switch ( 상태 ) {
    case 값1 :
        명령문 1-1;
        명령문 1-2;
    case 값2 :
        명령문 2-1;
        명령문 2-2;
    default :
        명령문 3-1;
        명령문 3-2;
}
```

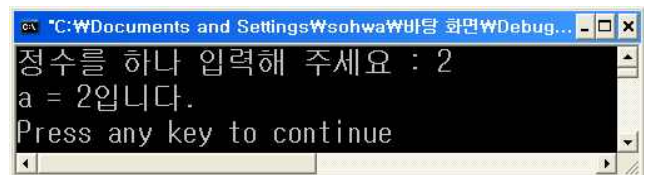
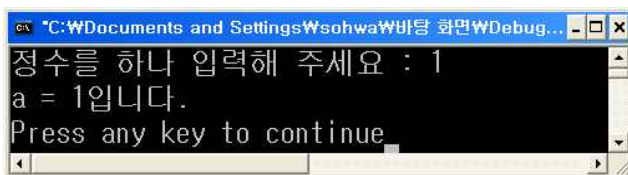


default는 case의 값 중 switch와 일치하는 값이 없을 경우에 실행되며 if문의 else와 같이 필요 없다면 생략 할 수 있다.

예제 4-12: switch ~ case문의 사용예제

```
#include <stdio.h>
void main ( ) {
    int a;
    printf("정수를 하나 입력해 주세요 : ");
    scanf("%d", &a);
    switch ( a ) {
        case 1 : printf("a = 1입니다.\n"); break;
        case 2 : printf("a = 2입니다.\n"); break;
    }
}
```

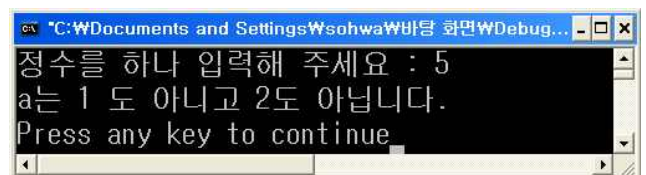
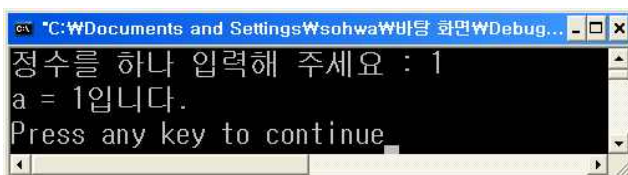
입력한 숫자별 실행 결과



예제 4-13: switch ~ case문에 default문을 추가한 예제

```
#include <stdio.h>
void main ( ) {
    int a;
    printf("정수를 하나 입력해 주세요 : ");
    scanf("%d", &a);
    switch ( a ) {
        case 1 : printf("a = 1입니다.\n"); break;
        case 2 : printf("a = 2입니다.\n"); break;
        default : printf("a는 1 도 아니고 2도 아닙니다.\n");
    }
}
```

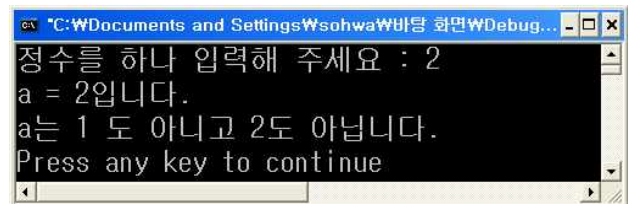
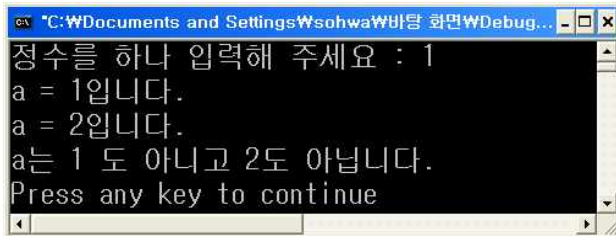
입력한 숫자별 실행 결과



예제 4-14: switch ~ case문에서 break가 없을 때의 실행 모양

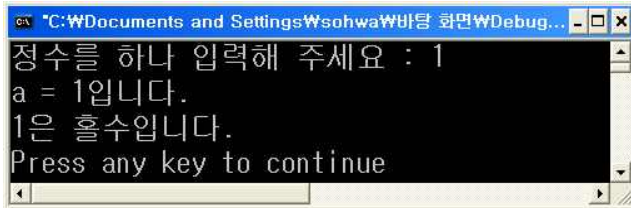
```
#include <stdio.h>
void main ( ) {
    int a;
    printf("정수를 하나 입력해 주세요 : ");
    scanf("%d", &a);
    switch ( a ) {
        case 1 : printf("a = 1입니다.\n");
        case 2 : printf("a = 2입니다.\n");
        default : printf("a는 1 도 아니고 2도 아닙니다.\n");
    }
}
```

입력한 숫자별 실행 결과

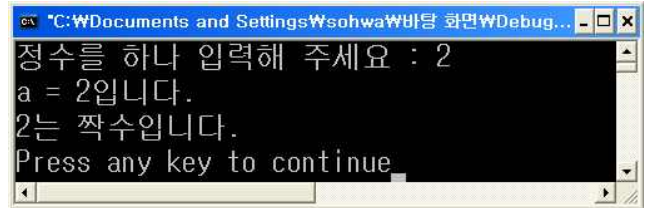


예제 4-15: switch ~ case문에서 명령 문장이 여러 개 있을 경우의 프로그램 방법

```
#include <stdio.h>
void main ( ) {
    int a;
    printf("정수를 하나 입력해 주세요 : ");
    scanf("%d", &a);
    switch ( a ) {
        case 1 :
            printf("a = 1입니다.\n");
            printf("1은 홀수입니다.\n");
            break;
        case 2 :
            printf("a = 2입니다.\n");
            printf("2는 짝수입니다.\n");
            break;
        default :
            printf("a는 1 도 아니고 2도 아닙니다.\n");
            printf("짝수인지 홀수 인지 모르겠군요.\n");
    }
}
```



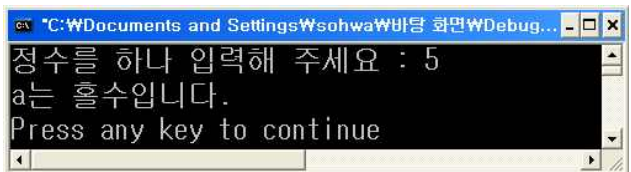
```
정수를 하나 입력해 주세요 : 1
a = 1입니다.
1은 홀수입니다.
Press any key to continue
```



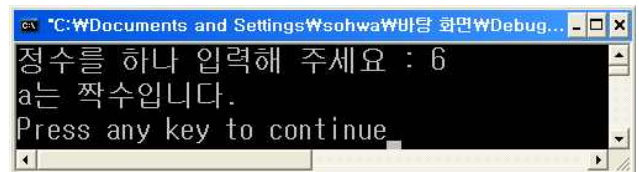
```
정수를 하나 입력해 주세요 : 2
a = 2입니다.
2는 짝수입니다.
Press any key to continue
```

예제 4-16: switch 뒤쪽에 수식을 넣은 프로그램

```
#include <stdio.h>
void main ( ) {
    int a;
    printf("정수를 하나 입력해 주세요 : ");
    scanf("%d", &a);
    switch ( a % 2 ) {
        case 0 : printf("a는 짝수입니다.\n"); break;
        case 1 : printf("a는 홀수입니다.\n"); break;
    }
}
```



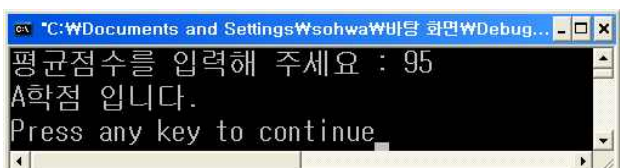
```
정수를 하나 입력해 주세요 : 5
a는 홀수입니다.
Press any key to continue
```



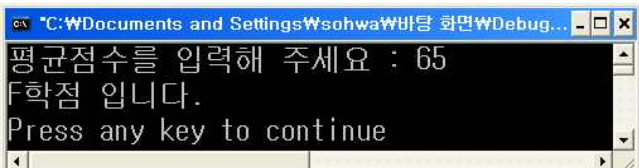
```
정수를 하나 입력해 주세요 : 6
a는 짝수입니다.
Press any key to continue
```

예제 4-17: switch ~ case 문을 이용한 학점 출력 프로그램

```
#include <stdio.h>
void main ( ) {
    double jumsu;
    printf("평균점수를 입력해 주세요 : ");
    scanf("%lf", &jumsu);
    switch ( jumsu / 10 ) {
        case 10 :
        case 9 : printf("A학점 입니다.\n"); break;
        case 8 : printf("B학점 입니다.\n"); break;
        case 7 : printf("C학점 입니다.\n"); break;
        default : printf("F학점 입니다.\n");
    }
}
```



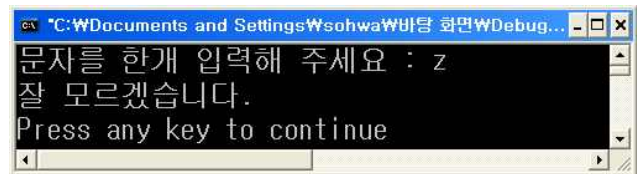
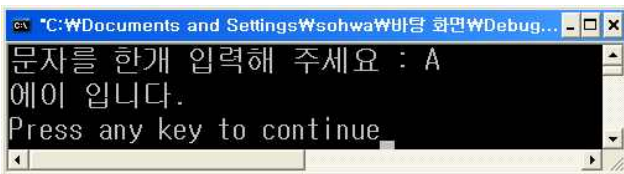
```
평균점수를 입력해 주세요 : 95
A학점 입니다.
Press any key to continue
```



```
평균점수를 입력해 주세요 : 65
F학점 입니다.
Press any key to continue
```

예제 4-18: char를 상태 값으로 하는 switch ~ case문

```
#include <stdio.h>
void main ( ) {
    char x;
    printf("문자를 한개 입력해 주세요 : ");
    scanf("%c", &x);
    switch ( x ) {
        case 'A' : printf("에이 입니다.\n");    break;
        case 'B' : printf("비  입니다.\n");    break;
        default  : printf("잘 모르겠습니다.\n");
    }
}
```



🔄 반복문

정해진 문장을 반복하여 실행할 수 있도록 해주는 제어문이다. 많은 양의 data를 효과적으로 처리 하는데 도움을 준다. C언어에는 for문, while문, do~ while문의 세 가지 반복문이 있다.

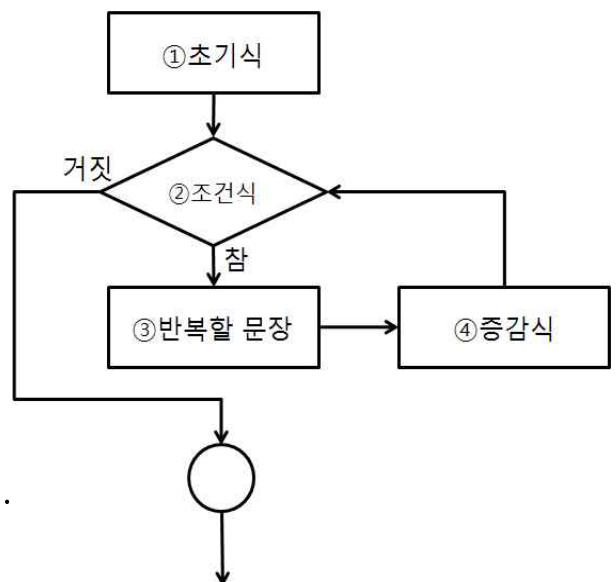
1. for 문

설명 : for문은 조건 반복문이라 부르며 작성 방법이 고정되어 있다. 사용법은 아래와 같다.

```
for ( ①초기식 ; ②조건식 ; ④증감식 ) {
    ③반복하려는 문장;
}
```

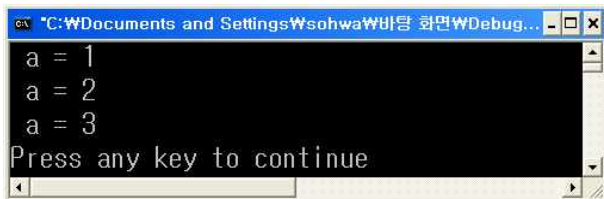
- ① 초기식은 한 번만 실행된다. 반복을 위해 변수의 값을 초기화 하고자 할 때 사용한다. 불필요 시 생략 가능하다.
- ② 조건식은 반복여부를 결정해 준다. 조건식이 참이면 문장이 실행된 후 증감식을 실행하고 다시 조건식을 본다. 조건식이 거짓이면 반복문은 종료된다. 조건식을 생략하면 조건은 항상 참이 된다.
- ③ 반복할 문장은 여러 개 쓸 수 있으며 여러 개의 문장을 반복하고자 하면 위처럼 해당 문장을 { }로 묶어야 한다.
- ④ 증감식은 조건식에 변화를 주기 위해 사용한다. 반복 후 조건식이 그대로면 무한으로 반복됨으로 조건식의 변화를 위해 증감식을 사용하며 불필요 시 생략 할 수 있다.

for문의 실행순서 : ①→②→③→④→②→③→④→...→②→조건이 거짓이면 종료 된다.



예제 4-19: for문의 사용 예제 - 1

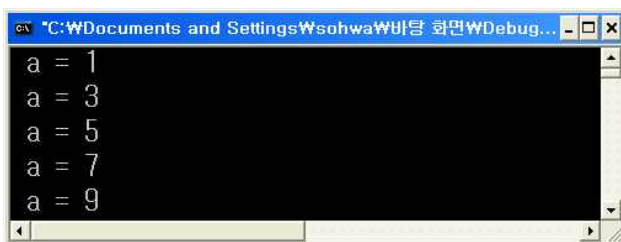
```
#include <stdio.h>
void main ( ) {
    int a;
    for ( a = 1 ; a <= 3 ; a++ ) {
        printf(" a = %d\n", a );
    }
}
```



```
C:\WDocuments and Settings\Wsohwa\바탕 화면\WDebug...
a = 1
a = 2
a = 3
Press any key to continue
```

예제 4-20: 증감식에서 a의 값을 2씩 증가 시키는 예제

```
#include <stdio.h>
void main ( ) {
    int a;
    for ( a = 1 ; a <= 10 ; a = a+2 ) {
        printf(" a = %d\n", a );
    }
}
```



```
C:\WDocuments and Settings\Wsohwa\바탕 화면\WDebug...
a = 1
a = 3
a = 5
a = 7
a = 9
```

예제 4-21: for문을 이용해 1~100까지의 수 중 짝수를 출력하는 프로그램 - 1

```
#include <stdio.h>
void main ( ) {
    int a;
    for ( a = 1 ; a <= 100 ; a = a++ ) {
        if( a % 2 == 0 ) {
            printf(" a = %d\n", a );
        }
    }
}
```

예제 4-22: for문을 이용해 1~100까지의 수 중 짝수를 출력하는 프로그램 - 2

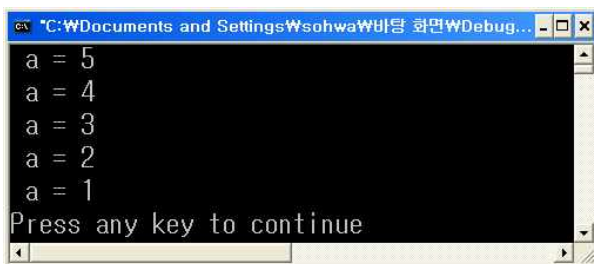
```
#include <stdio.h>
void main ( ) {
    int a;
    for ( a = 2 ; a <= 100 ; a = a+2 ) {
        printf(" a = %d\n", a );
    }
}
```

예제 4-23: for문을 이용해 1~100까지의 수 중 짝수를 출력하는 프로그램 - 3

```
#include <stdio.h>
void main ( ) {
    int a;
    for ( a = 1 ; a <= 50 ; a = a++ ) {
        printf(" a = %d\n", a*2 );
    }
}
```

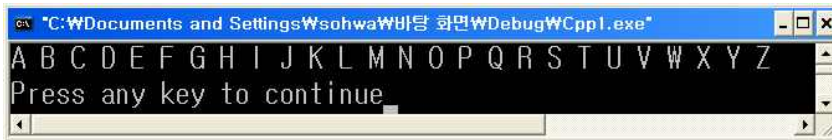
예제 4-24: 변수의 값을 1씩 감소시키며 출력하는 예제

```
#include <stdio.h>
void main ( ) {
    int a;
    for ( a = 5 ; a > 0 ; a = a-- ) {
        printf(" a = %d\n", a );
    }
}
```



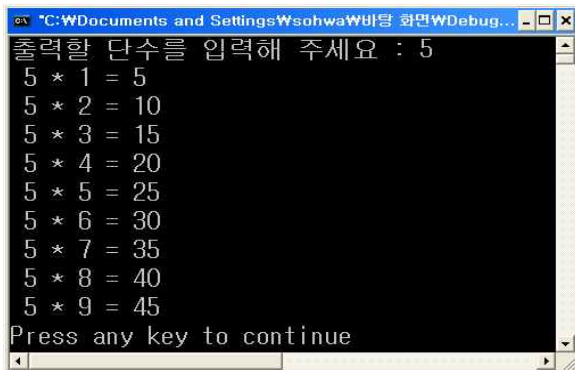
예제 4-25: 반복문을 이용해 알파벳을 출력한 예제

```
#include <stdio.h>
void main ( ) {
    char a;
    for ( a = 'A' ; a <= 'Z' ; a++ ) {
        printf("%c ", a );
    }
    printf("\n");
}
```

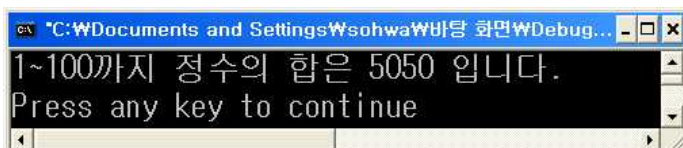
예제 4-26: 정수를 입력받아 구구단 중 입력받은 정수에 해당하는 단을 출력하는 예제

```
#include <stdio.h>
void main ( ) {
    int dan, a;
    printf("출력할 단수를 입력해 주세요 : " );
    scanf("%d", &dan);
    for ( a = 1 ; a <= 9 ; a++ ) {
        printf(" %d * %d = %d Wn", dan, a, dan*a );
    }
}
```



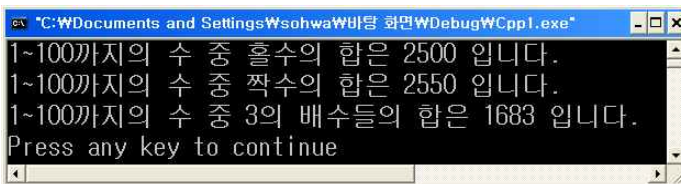
예제 4-27: for문을 이용해 1부터 100까지의 합을 구하는 프로그램

```
#include <stdio.h>
void main ( ) {
    int a, sum = 0;
    for ( a = 1 ; a <= 100 ; a++ ) {
        sum = sum + a;
    }
    printf("1~100까지 정수의 합은 %d 입니다.Wn", sum);
}
```



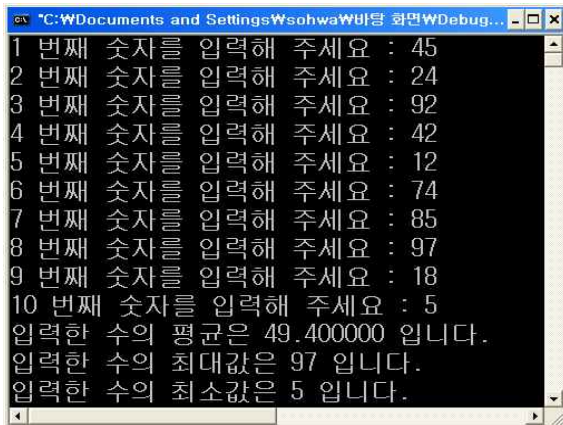
예제 4-28: for 문을 이용해 1부터 100까지의 정수 중 짝수와 홀수 그리고 3의 배수들의 합을 구하는 프로그램

```
#include <stdio.h>
void main ( ) {
    int a, even = 0, odd = 0, sam = 0;
    for ( a = 1 ; a <= 100 ; a++ ) {
        if( a %2 != 0 ) {
            even = even + a;
        }
        else {
            odd = odd + a;
        }
        if( a % 3 == 0 ) {
            sam = sam + a;
        }
    }
    printf("1~100까지의 수 중 홀수의 합은 %d 입니다.\n", even);
    printf("1~100까지의 수 중 짝수의 합은 %d 입니다.\n", odd);
    printf("1~100까지의 수 중 3의 배수들의 합은 %d 입니다.\n", sam);
}
```



예제 4-29: 10개의 점수를 입력 받아 입력받은 점수의 평균과 최대값 그리고 최소값을 구하는 예제. 단, 이때 점수는 0 ~ 100점 사이에 있다고 가정한다.

```
#include <stdio.h>
void main ( ) {
    int in, sum = 0, max = 0, min = 100, a;
    for ( a = 1 ; a <= 10 ; a++ ) {
        printf("%d 번째 숫자를 입력해 주세요 : ", a);
        scanf("%d", &in);
        sum = sum + in;                // 총점을 구하는 방법
        if( max < in ) {                // 최대값을 구하는 방법
            max = in;
        }
        if( min > in ) {                // 최소값을 구하는 방법
            min = in;
        }
    }
    printf("입력한 수의 평균은 %lf 입니다.\n", sum/10.0);
    printf("입력한 수의 최대값은 %d 입니다.\n", max);
    printf("입력한 수의 최소값은 %d 입니다.\n", min);
}
```



```
C:\Documents and Settings\Wsohwa\바탕 화면\WDebug...
1 번째 숫자를 입력해 주세요 : 45
2 번째 숫자를 입력해 주세요 : 24
3 번째 숫자를 입력해 주세요 : 92
4 번째 숫자를 입력해 주세요 : 42
5 번째 숫자를 입력해 주세요 : 12
6 번째 숫자를 입력해 주세요 : 74
7 번째 숫자를 입력해 주세요 : 85
8 번째 숫자를 입력해 주세요 : 97
9 번째 숫자를 입력해 주세요 : 18
10 번째 숫자를 입력해 주세요 : 5
입력한 수의 평균은 49.400000 입니다.
입력한 수의 최대값은 97 입니다.
입력한 수의 최소값은 5 입니다.
```

예제 4-30: 초기식을 생략한 반복문 예제

```
#include <stdio.h>
void main ( ) {
    int a = 1;
    for ( ; a <= 3 ; a++ ) { // a가 이미 초기화 되어 있으므로 초기화 할 필요가 없다.
        printf("a = %d \n", a);
    }
}
```

예제 4-31: 조건식을 생략한 반복문 예제

```
#include <stdio.h>
void main ( ) {
    int a = 1;
    for ( ; ; a++ ) { // 조건식을 생략하면 조건이 항상 참이 되어 무한 반복된다.
        printf("a = %d \n", a);
    }
}
```

예제 4-32: 증감식을 생략한 반복문 예제 - 1

```
#include <stdio.h>
void main ( ) {
    int a = 1;
    for ( ; a <= 3 ; ) { // 증감식에 의한 조건식의 변화가 없다면 무한 반복된다.
        printf("a = %d \n", a);
    }
}
```

예제 4-33: 증감식을 생략한 반복문 예제 - 2

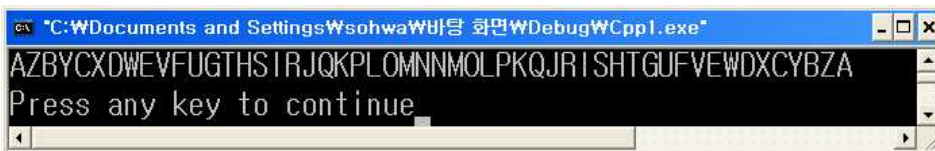
```
#include <stdio.h>
void main ( ) {
    int a = 1;
    for ( ; a <= 3 ; ) {
        printf("a = %d \n", a++); // 내용에 증감식이 있으므로 별도의 증감식을 만들
    } // 필요가 없다.
}
```

예제 4-34: 초기식, 조건식, 증감식을 모두 생략한 반복문 예제

```
#include <stdio.h>
void main ( ) {
    int a = 1;
    for ( ; ; ) {                // 조건이 참임으로 무한 반복 된다.
        printf("a = %d \n", a++);
    }
}
```

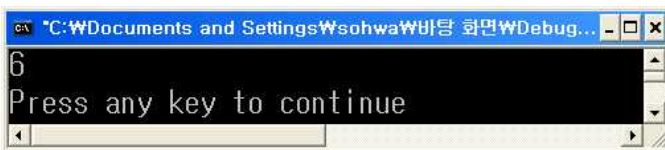
예제 4-35: 초기식과 증감식을 두 개씩 사용한 반복문 예제

```
#include <stdio.h>
void main ( ) {
    char a, b;
    for ( a = 'A', b = 'Z' ; a <= 'Z' ; a++, b-- ) {
        printf("%c%c", a, b);
    }
    printf("\n");
}
```



예제 4-36: for 문의 뒤쪽에 ;을 썼을 경우 발생하는 문제점

```
#include <stdio.h>
void main ( ) {
    int a;
    for ( a = 1; a <= 5 ; a++ ); {
        printf("%d ", a);
    }
    printf("\n");
}
```



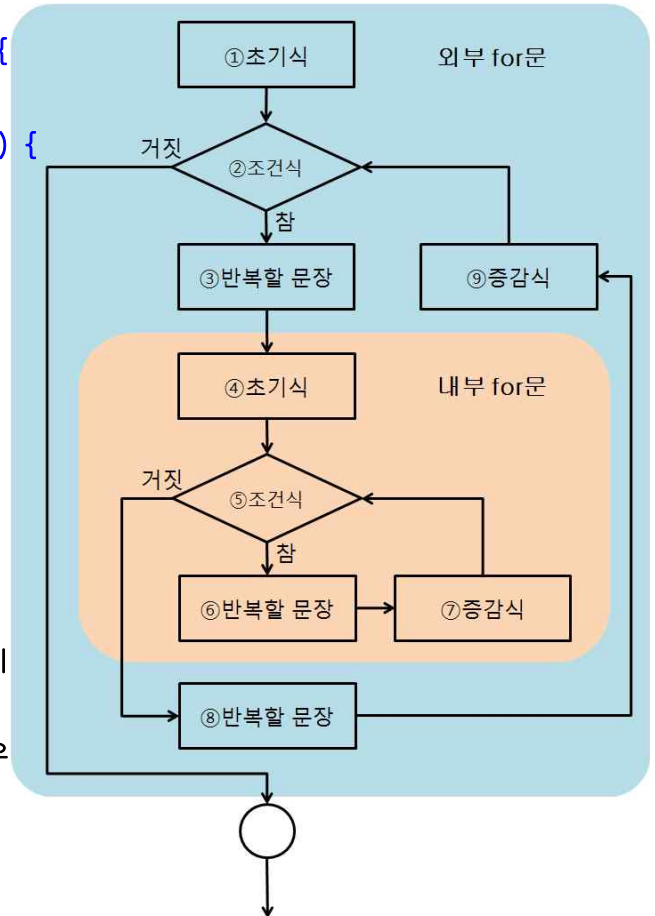
설명 : for뒤쪽에 세미콜론을 입력했기 때문에 for 문은 자신이 반복시켜야할 문장이 없다고 판단해 아무 문장도 실행하지 않고 단순히 반복만 하게 된다. 그렇기 때문에 printf를 이용해 a를 출력하면 6이 나오는 것이다.

2. 다중 for 문

설명 : for문안에 있는 반복 대상문장이 for문인 경우를 다중 반복문이라 한다.

```
for ( ①초기식 ; ②조건식 ; ⑨증감식 ) {
    ③반복하려는 문장 1;
    for( ④초기식 ; ⑤조건식; ⑦증감식 ) {
        ⑥반복하려는 문장 2;
    }
    ⑧반복하려는 문장 3;
}
```

②의 조건식이 참이라면 ③이 실행된 후 내부 반복문이 동작한다. 내부에 있는 반복문이 종료 되면 ⑧명령을 수행하고 ⑨의 증감식을 실행한 뒤 ②의 조건식으로 다시 이동해 조건을 비교한다. 즉, 이 흐름도는 하나의 반복문이 반복하는 문장이 ③, 반복문, ⑧의 세 개인 반복문에 대한 흐름도이다. 이와같은 반복문을 만능 경우 프로그램의 동작 순서는 다음과 같다.



for문의 실행순서 :

①->②->③->④->⑤->⑥->⑦->⑤->⑥->⑦.....->⑤->⑥->⑦->⑤가 거짓이라면->⑧->⑨->
 ②->③->④->⑤->⑥->⑦->⑤->⑥->⑦.....->⑤->⑥->⑦->⑤가 거짓이라면->⑧->⑨->
 ②->③->④->⑤->⑥->⑦->⑤->⑥->⑦.....->⑤->⑥->⑦->⑤가 거짓이라면->⑧->⑨->
 ②가 거짓이라면 종료.

예제 4-37: 이중반복문에 대한 이해 예제 - 1

```
#include <stdio.h>
void main ( ) {
    int i, j;
    for ( i = 1; i <= 2 ; i++ ) {
        for ( j = 1; j <= 3 ; j++ ) {
            printf("i= %d, j=%d\n", i, j);
        }
    }
}
```

```

i= 1, j=1
i= 1, j=2
i= 1, j=3
i= 2, j=1
i= 2, j=2
i= 2, j=3
Press any key to continue

```

예제 4-39: 이중반복문에 대한 이해 예제 - 2

```

#include <stdio.h>
void main ( ) {
    int i, j, k=1;
    for ( i = 1; i <= 2 ; i++ ) {
        for ( j = 1; j <= 3 ; j++ ) {
            printf("i= %d, j=%d, k=%d\n", i, j, k++);
        }
    }
}

```

```

i= 1, j=1, k=1
i= 1, j=2, k=2
i= 1, j=3, k=3
i= 2, j=1, k=4
i= 2, j=2, k=5
i= 2, j=3, k=6
Press any key to continue

```

예제 4-40: 이중반복문에 대한 이해 예제 - 3

```

#include <stdio.h>
void main ( ) {
    int i, j;
    for ( i = 1; i <= 2 ; i++ ) {
        printf("외부 for문이 %d번째 턴이 시작됩니다.\n", i);
        for ( j = 1; j <= 3 ; j++ ) {
            printf("내부 for문의 %d턴입니다. \n", j );
        }
        printf("외부 for문의 %d번째 턴이 종료합니다.\n", i);
    }
}

```

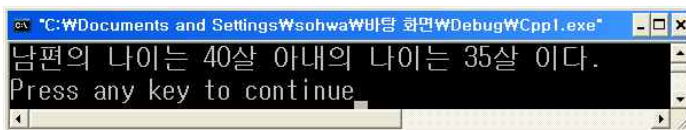
```

외부 for문이 1번째 턴이 시작됩니다.
내부 for문의 1턴입니다.
내부 for문의 2턴입니다.
내부 for문의 3턴입니다.
외부 for문의 1번째 턴이 종료합니다.
외부 for문이 2번째 턴이 시작됩니다.
내부 for문의 1턴입니다.
내부 for문의 2턴입니다.
내부 for문의 3턴입니다.
외부 for문의 2번째 턴이 종료합니다.
Press any key to continue

```

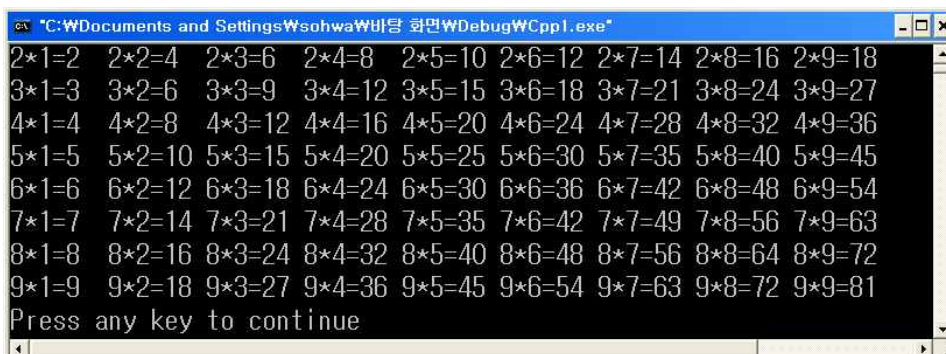
예제 4-41: 이중 반복문을 이용한 일차방정식의 풀이에 대한 예제. 남편과 아내의 나이의 합은 75살이고 나이의 곱은 1400이다. 아내보다 남편의 나이가 더 많다고 할 때 남편과 아내의 나이는 각 몇 살인지 찾아내는 프로그램을 작성해라. 단, 남편과 아내의 나이는 100살을 넘지 않는다.

```
#include <stdio.h>
void main ( ) {
    int x, y;
    for ( x = 1; x <= 100 ; x++ ) {
        for ( y = 1; y <= 100 ; y++ ) {
            if( x+y==75 && x*y==1400 && x>y ) {
                printf("남편의 나이는 %d살 아내의 나이는 %d살 이다.\n", x, y);
            }
        }
    }
}
```



예제 4-42: 이중 반복문을 이용한 구구단의 출력 방법 (가로로 구구단 출력)

```
#include <stdio.h>
void main ( ) {
    int x, y;
    for ( x = 2; x <= 9 ; x++ ) {
        for ( y = 1; y <= 9 ; y++ ) {
            printf("%d*%d=%-3d", x, y, x*y);
        }
        printf("\n");
    }
}
```



예제 4-43: 이중 반복문을 이용한 구구단의 출력 방법 (세로로 구구단 출력)

```
#include <stdio.h>
void main ( ) {
    int x, y;
    for ( x = 1; x <= 9 ; x++ ) {
        for ( y = 2; y <= 9 ; y++ ) {
            printf("%d*%d=%-3d", y, x, x*y);
        }
        printf("\n");
    }
}
```

```
2*1=2 3*1=3 4*1=4 5*1=5 6*1=6 7*1=7 8*1=8 9*1=9
2*2=4 3*2=6 4*2=8 5*2=10 6*2=12 7*2=14 8*2=16 9*2=18
2*3=6 3*3=9 4*3=12 5*3=15 6*3=18 7*3=21 8*3=24 9*3=27
2*4=8 3*4=12 4*4=16 5*4=20 6*4=24 7*4=28 8*4=32 9*4=36
2*5=10 3*5=15 4*5=20 5*5=25 6*5=30 7*5=35 8*5=40 9*5=45
2*6=12 3*6=18 4*6=24 5*6=30 6*6=36 7*6=42 8*6=48 9*6=54
2*7=14 3*7=21 4*7=28 5*7=35 6*7=42 7*7=49 8*7=56 9*7=63
2*8=16 3*8=24 4*8=32 5*8=40 6*8=48 7*8=56 8*8=64 9*8=72
2*9=18 3*9=27 4*9=36 5*9=45 6*9=54 7*9=63 8*9=72 9*9=81
Press any key to continue
```

예제 4-44: 이중 반복문을 이용한 별의 출력

```
#include <stdio.h>
void main ( ) {
    int x, y;
    for ( x = 1; x <= 5 ; x++ ) {
        for ( y = 1; y <= 5 ; y++ ) {
            printf("□");
        }
        printf("\n");
    }
}
```

```
□□□□□
□□□□□
□□□□□
□□□□□
□□□□□
Press any key to continue
```

3. while 문

설명 : for문과 동일하게 반복을 위해 사용하는 문장이지만 for처럼 초기식, 조건식, 증감식 기재 부분이 정형화 되어있지 않기 때문에 자유로운 형태로 원하는 내용은 반복 할 수 있다.

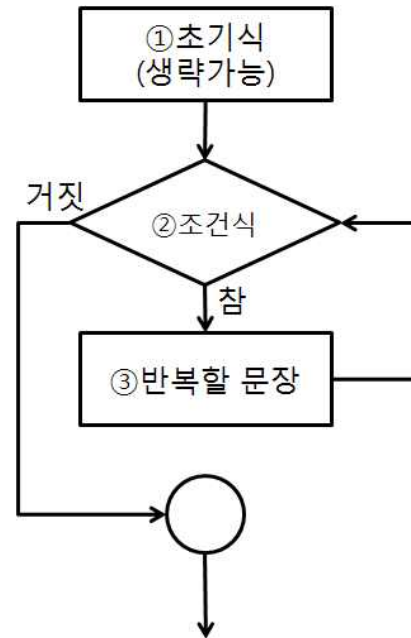
①초기식

while(②조건식) {

③명령문

}

while문은 자신의 ②조건식이 만족한다면 아래쪽의 ③명령문을 실행하고 그렇지 않는다면 종료한다. 따라서 증감식 없는 for문과 유사하기 때문에 ③명령문 내에 증감식을 넣어 조건식에 대한 변화를 주는 형태로 사용한다. 또한 while문은 for문과 다르게 조건식을 생략 할 수 없다. while문의 조건은 반드시 참 또는 거짓으로 결과가 나와야 하며 for문처럼 조건식을 항상 참으로 나타내고 싶다면 조건에 1을 넣는다.



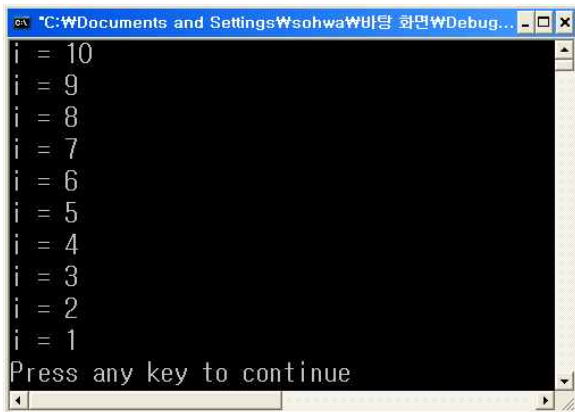
예제 4-45: while을 이용해 1~10까지 출력하는 예제

```
#include <stdio.h>
void main ( ) {
    int i=1;                // 초기식
    while( i<=10 ) {        // 조건식
        printf("i = %d\n", i++); // 명령문과 증감식
    }
}
```

```
i = 1
i = 2
i = 3
i = 4
i = 5
i = 6
i = 7
i = 8
i = 9
i = 10
Press any key to continue
```

예제 4-46: while을 이용해 10부터 1까지 출력하는 예제

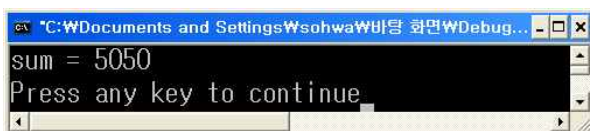
```
#include <stdio.h>
void main ( ) {
    int i=10;                // 초기식
    while( i ) {             // 조건식
        printf("i = %d\n", i--); // 명령문과 증감식
    }
}
```



```
C:\Documents and Settings\sohwa\바탕 화면\WDebug...
i = 10
i = 9
i = 8
i = 7
i = 6
i = 5
i = 4
i = 3
i = 2
i = 1
Press any key to continue
```

예제 4-46: while을 이용해 1부터 100까지 합을 출력하는 예제

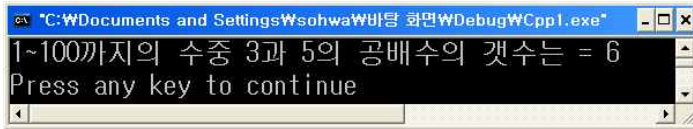
```
#include <stdio.h>
void main ( ) {
    int i = 1, sum = 0;      // 초기식
    while( i<=100 ) {       // 조건식
        sum = sum + i++;    // 명령문과 증감식
    }
    printf("sum = %d\n", sum);
}
```



```
C:\Documents and Settings\sohwa\바탕 화면\WDebug...
sum = 5050
Press any key to continue
```

예제 4-47: while을 이용한 1부터 100까지의 수 중 3과 5의 공배수의 개수를 출력하는 예제

```
#include <stdio.h>
void main ( ) {
    int i = 1, su = 0;      // 초기식
    while( i<=100 ) {      // 조건식
        if( i%3==0 && i%5==0 )
            su++;
        i++;
    }
    printf("1~100까지의 수중 3과 5의 공배수의 갯수는 = %d\n", su);
}
```



4. do~ while 문

설명 : while문과 사용방법이 동일하지만 조건식을 비교하기 전 먼저 do뒤쪽의 명령문을 실행하고 조건을 판단한다. 따라서 조건과 관계없이 명령문은 한번 실행됨으로 어떠한 문장은 무조건 한번은 실행해야하는 반복문을 만들때는 do~ while문을 사용한다.

①초기식

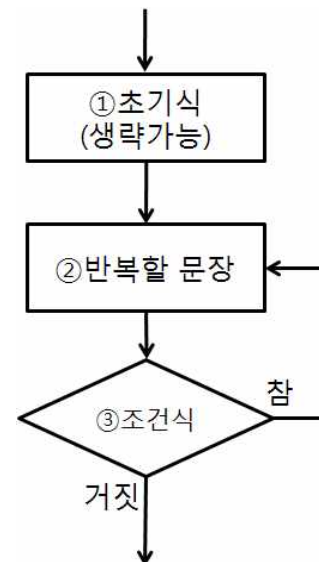
do {

②명령문

}

while(③조건식);

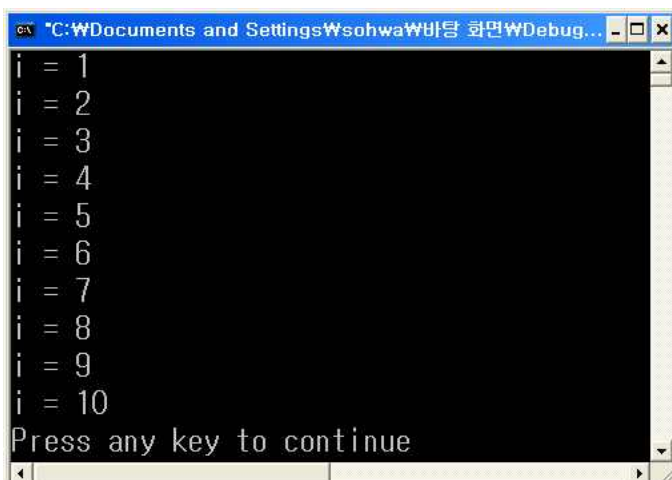
do ~ while문은 자신의 ②명령문을 우선 한번 실행 한 뒤 ③조건식이 만족한다면 ②명령문을 반복하여 실행하고 그렇지 않는다면 종료한다. 따라서 조건의 뒤쪽에 반드시 ;을 써주어야 하며 초기식, 증감식 등의 입력 방법은 모두 while문과 동일하다.



예제 4-48: do~ while을 이용해 1부터 10까지의 수를 출력하는 예제

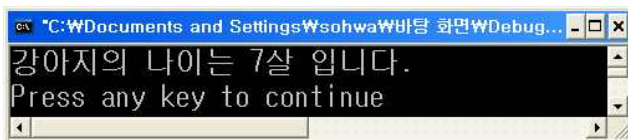
```

#include <stdio.h>
void main ( ) {
    int i = 1;                // 초기식
    do {
        printf("i = %d\n", i++);
    }while( i<=10 );
}
  
```



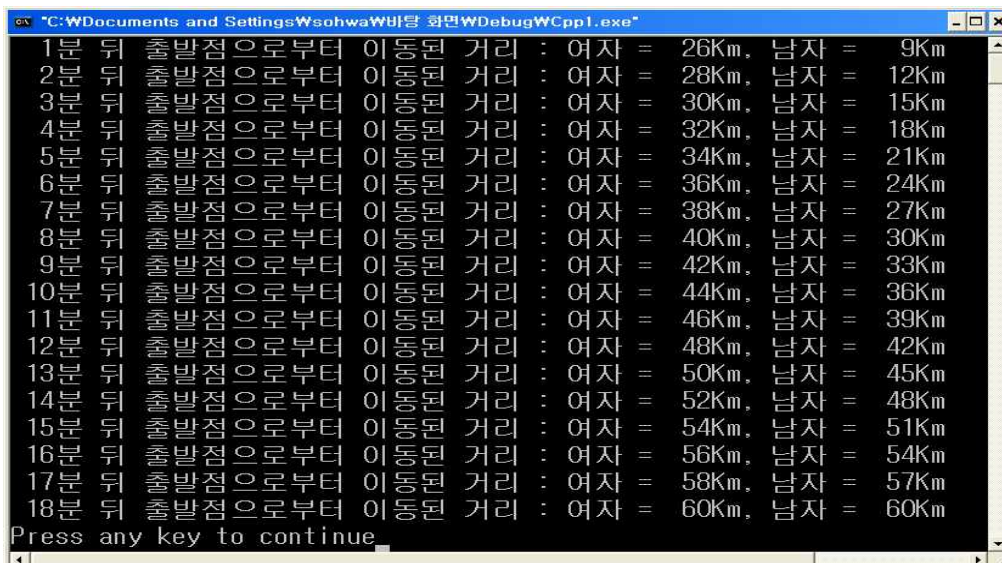
예제 4-49: do~ while을 이용한 방정식의 풀이 예제 - 1. 남자의 나이는 38살이다. 남자의 집에는 강아지가 있는데 남자의 나이는 강아지의 나이에 5배보다 3살이 더 많다. 강아지의 나이는 몇 살인지 프로그램을 통해 구해 보아라.

```
#include <stdio.h>
void main ( ) {
    int x = 1;                // 초기식
    do {
        x++;
    }while( x*5+3 != 38 );
    printf("강아지의 나이는 %d살 입니다.\n", x);
}
```



예제 4-50: do~ while을 이용한 방정식의 풀이 예제 - 2. 여자의 자동차는 시속 120Km으로 달리고 있고 남자의 자동차는 시속 180Km로 달리고 있다. 여자의 자동차는 현재 출발지점에서 24Km를 진행한 상태이고 남자는 출발지점에서 6Km를 진행한 상태이다. 남자는 몇 분 뒤에 여자의 자동차를 추월하게 되는지를 알아내는 프로그램을 작성하여라.

```
#include <stdio.h>
void main ( ) {
    int x = 0;
    int man, woman;
    do {
        x++;
        woman = 24 + x * 2;
        man = 6 + x * 3;
        printf("%3d분 뒤 출발점으로부터 이동된 거리 : ", x);
        printf("여자 = %3dKm, 남자 = %3dKm\n", woman, man);
    }while( man < woman );
}
```



🔄 분기문

프래그램의 제어권을 강제로 변경 할 수 있는 제어문 들을 말한다. C언어에는 크게 break문, continue문, goto문, return문 등이 있다.

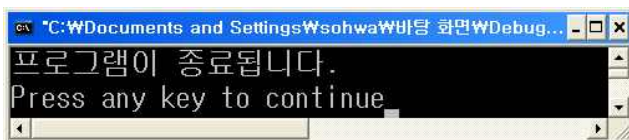
1. break 문

설명 : 원하는 반복문의 실행을 멈추거나 switch case문의 명령 실행을 종료하기 위해 사용된다. 반복문 안에서 break문을 사용하면 해당 반복문은 즉시 정지된다. 단, break문은 한 번에 한 개의 반복문만 탈출 할 수 있다. 또한 반복문이 없는 곳에서 break문을 사용하면 에러가 발생하게 된다.

- ① 반복문의 동작을 멈추기 위해 사용 한다. 따라서 반복문 안에 사용하지 않으면 에러가 발생한다. (단, switch 문안에 사용하는 건 예외로 한다.)
- ② 반복문 안에서 break가 항상 실행되면 반복문 자체가 의미가 없어짐으로 break문은 일반적으로 if문과 함께 사용해 특정 조건을 만족 했을 경우에만 실행되게 한다.

예제 4-50: break문을 이용해 반복문을 강제로 탈출하는 예제

```
#include <stdio.h>
void main ( ) {
    int a;
    for( a = 1; a <=1000000000; a++) {
        break;
        printf("a = %d\n", a);
    }
    while( 1 ) {
        break;
        printf("a = %d\n", a);
    }
    do {
        break;
        printf("a = %d\n", a);
    }while(1);
    printf("프로그램이 종료됩니다.\n");
}
```



예제 4-51: break문의 활용 예 - 1

```
#include <stdio.h>
void main ( ) {
    int a = 1;
    for( ; ; a++ ) {
        printf("a = %d, ", a);
        if( a == 10 ) {
            break;
        }
    }
    printf("\nfor 문 종료\n");
    while( 1 ) {
        printf("a = %d, ", a--);
        if( !a ) {
            break;
        }
    }
    printf("\nwhile 문 종료\n");
    do {
        printf("a = %d, ", ++a);
        if( a == 10 ) {
            break;
        }
    }while(1);
    printf("\ndo ~ while문 종료.\n");
}
```

예제 4-51: 숫자 3을 입력 받을 때 까지 반복하는 프로그램

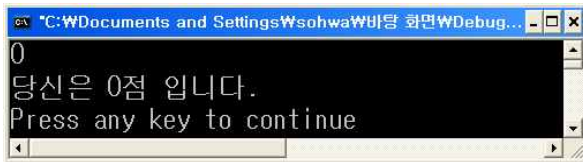
```
#include <stdio.h>
void main ( ) {
    int a;
    while( 1 ) {
        printf("나와 싸우려면 아무숫자나 입력하고 항복하려면 3을 입력 하세요 : ");
        scanf("%d", &a);
        if ( a == 3 ) {
            break;
        }
        printf("다시한번 생각해 보세요\n");
    }
    printf("결국 항복 했군요\n");
}
```


예제 4-52: 원하는 범위의 수가 입력될 때까지 반복해 입력 받는 프로그램

```
#include <stdio.h>
void main ( ) {
    int nai;
    while( 1 ) {
        printf("나이를 입력해 주세요 : ");
        scanf("%d", &nai);
        if ( nai < 0 ) {
            printf("나이는 영보다 작을 수 없습니다. 다시입력 하세요. Wn");
        }
        else if ( nai > 150 ) {
            printf("거짓말 마세요. 다시 입력 하세요. Wn");
        }
        else {
            break;
        }
    }
    printf("당신의 나이는 %d살 입니다.Wn", nai);
}
```

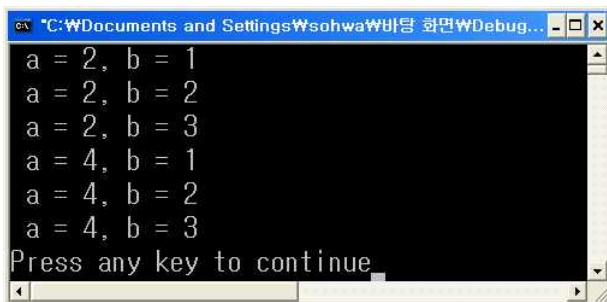
예제 4-53: 반복문을 이용한 간단한 게임만들기.

```
#include <stdio.h>
#include <conio.h>
void main ( ) {
    int num, a = 1;
    while( 1 ) {
        printf("%d", a);
        if( kbhit( ) ) {
            break;
        }
        a = (a+1) % 10;
        printf("Wb");
    }
    printf("Wn당신은 %d점 입니다.Wn", a);
    getch();
}
```



예제 4-54: break문은 한 번에 하나의 반복문만 탈출한다는 것을 보여주는 예제

```
#include <stdio.h>
void main ( ) {
    int a, b;
    for( a = 1; a <= 5 ; a++) {
        for( b = 1; b <=3; b++) {
            if( a % 2 != 0 ) {
                break;
            }
            printf(" a = %d, b = %d\n", a, b);
        }
    }
}
```



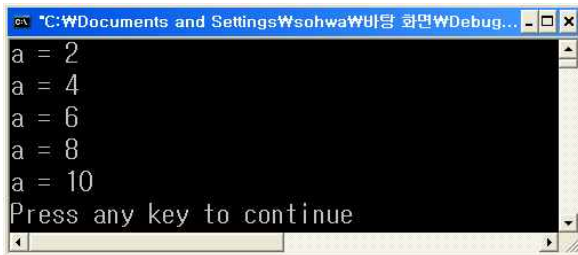
2. continue 문

설명 : 반복문이 실행해야할 명령 중 아직 실행하지 않은 명령들을 무시하고 바로 다음 단계인 조건식 또는 증감식으로 이동하게 해주는 제어문이다.

- ① 반복문의 제어권을 강제 변경하기 위해 사용 한다. 따라서 반복문 안에 사용하지 않으면 에러가 발생한다.
- ② continue가 실행되면 for문 같은 경우에는 증감식으로, while이나 do~ while문 같은 경우에는 조건식으로 제어권이 강제 이동된다.
- ③ continue도 break와 유사하게 항상 실행되면 continue 아래쪽의 명령문이 사용되지 못한다는 뜻이 됨으로 항상 if문과 같이 사용한다.

예제 4-55: continue 문을 사용해 1부터 10까지의 수 중 짝수만 출력하는 방법

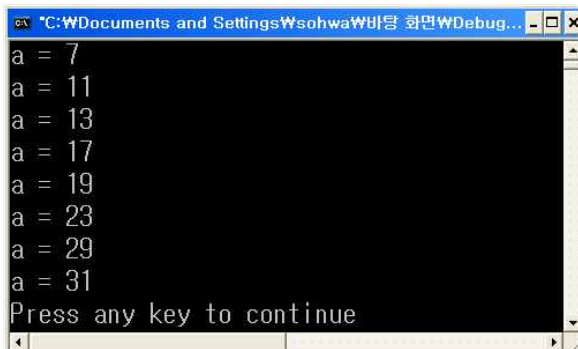
```
#include <stdio.h>
void main ( ) {
    int a;
    for( a = 1; a <= 10 ; a++ ) {
        if( a%2 != 0 ) {
            continue;
        }
        printf("a = %d\n", a);
    }
}
```



```
C:\Documents and Settings\Wsohwa\바탕 화면\WDebug...
a = 2
a = 4
a = 6
a = 8
a = 10
Press any key to continue
```

예제 4-56: continue 문을 사용해 1부터 30까지의 수 중 2의 배수, 3의 배수, 5의 배수가 아닌 수만 출력하는 방법

```
#include <stdio.h>
void main ( ) {
    int a = 1;
    while( a++ <=30 ) {
        if( a%2 == 0 || a%3 == 0 || a%5 == 0 ) {
            continue;
        }
        printf("a = %d\n", a);
    }
}
```



```
C:\Documents and Settings\Wsohwa\바탕 화면\WDebug...
a = 7
a = 11
a = 13
a = 17
a = 19
a = 23
a = 29
a = 31
Press any key to continue
```

3. goto 문

설명 : 프로그램 내의 지정한 곳으로 제어권을 강제 이동시켜주는 제어문

명칭1 :

명령문 1;

if(조건)

goto 명칭1;

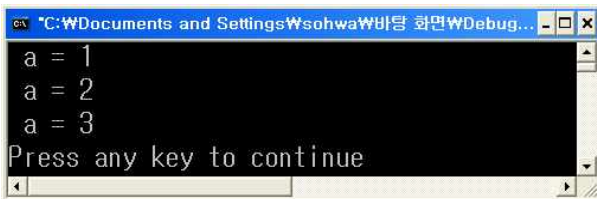
명령문 2;

① 지정한 곳으로 강제 이동이 가능하다.

② 프로그램이 복잡해짐으로 가능하면 사용하지 않는 것이 좋다.

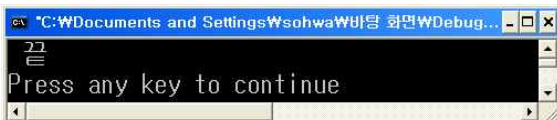
예제 4-57: goto 문은 지정한 곳으로 강제 이동 할 수 있다.

```
#include <stdio.h>
void main ( ) {
    int a = 1;
xx:
    printf(" a = %d\n", a++);
    if( a <= 3 ) {
        goto xx;
    }
}
```



예제 4-58: goto 문은 지정한 프로그램 후반부로도 강제 이동 할 수 있다.

```
#include <stdio.h>
void main ( ) {
    int a = 1;
    goto xx;
    for( a = 1 ; a <= 10 ; a++) {
        printf(" a = %d\n", a);
    }
xx:
    printf(" 끝\n");
}
```



4. return 문

설명 : 함수 내에서 사용하면 해당 문장이 실행된 함수가 강제 종료된다. main 함수 내에서 return 문을 실행하면 main 함수가 종료됨으로 프로그램이 종료되게 된다.

- ① 원하는 함수를 강제 종료 할 수 있다.
- ② 함수가 종료되며 원하는 값을 함수를 호출한 곳으로 돌려보낼 수 있다.
(함수의 리턴, 다음 챕터에서 자세하게 설명 함)