

프로그래밍을 통한 논리적 사유 연습





9장. 반복 문

(goto, break, continue)



goto문



- for문, while문 그리고 do~while문 이외에도 프로그램의 흐름을 변경할 때는 goto문, break문, continue문, return문 등을 사용한다.
- goto문
 - 실행 순서를 사용자가 지정한 라벨로 이동시킴
 - 명령문의 실행 순서를 프로그래머가 임의로 변경하고자 할 때 사용함
 - goto문은 실행 순서를 지정된 라벨로 옮겨 프로그램을 계속 수행함
- 기본형식

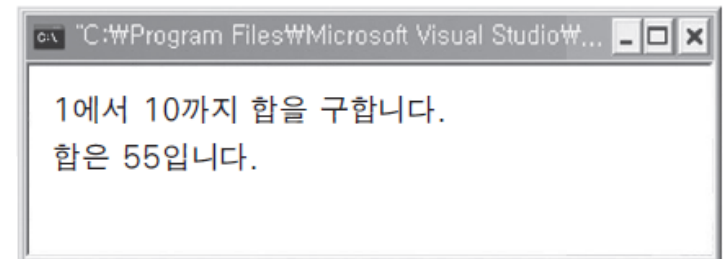
```
goto 라벨(Label);
```

- 라벨(Label)은 프로그램의 다른 명령문 앞에 선언함

goto문

예제) goto문을 사용하여 반복을 중단

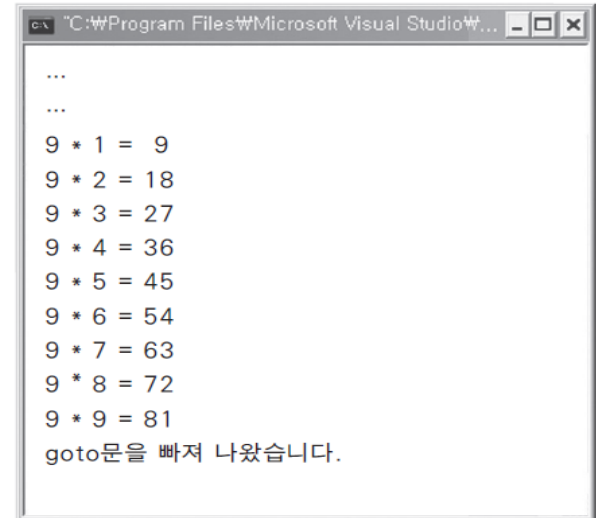
```
01 #include <stdio.h>
02
03 int main(void)
04 {
05     int sum = 0;
06     int number = 0;
07
08     while(1)
09     {
10         sum += number;
11         number++;
12
13         if(number > 10) goto ex1;
14     }
15
16     ex1:
17     printf("1부터 10까지 합을 구합니다.\n");
18     printf("합은 %d입니다.\n", sum);
19
20     return 0;
21 }
```



goto문

예제) 구구단을 중단

```
01 #include <stdio.h>
02 int main(void)
03 {
04     int i, j;
05
06     for(i = 1; i <= 100; i++)
07     {
08         for(j = 1; j <= 9; j++)
09         {
10             printf("%d * %d = %2d\n", i, j, i*j);
11             if(i == 9 && j == 9) goto end;
12         }
13     }
14
15     end:
16     printf("goto문을 빠져 나왔습니다.\n");
17
18     return 0;
19 }
```



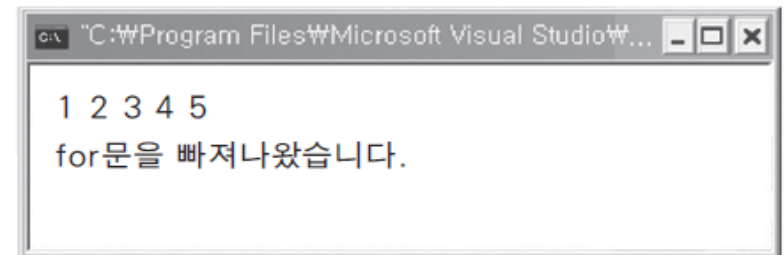
```
C:\Program Files\Microsoft Visual Studio\...
...
...
9 * 1 = 9
9 * 2 = 18
9 * 3 = 27
9 * 4 = 36
9 * 5 = 45
9 * 6 = 54
9 * 7 = 63
9 * 8 = 72
9 * 9 = 81
goto문을 빠져 나왔습니다.
```

break문

- break문
 - 반복문을 빠져나가거나 switch~case문에서 정상적인 흐름을 변경시킴

예제) 간단한 break문 사용

```
01 #include <stdio.h>
02
03 int main(void)
04 {
05     int i;
06
07     for(i = 1; i < 100; i++)
08     {
09         printf("%d ", i);
10         if(i >= 5) break;
11     }
12
13     printf("\n for문을 빠져나왔습니다.\n");
14
15     return 0;
16 }
```



```
C:\Program Files\Microsoft Visual Studio\...
1 2 3 4 5
for문을 빠져나왔습니다.
```

continue문

- continue문
 - 반복 문을 빠져 나오지 않고, 해당 반복문의 처음으로 흐름을 옮김
- 1~10에서 홀수만 더하는 예 (1)

```
int sum = 0, i;  
  
for(i=1; i<=10; i+=2)          // 초깃값이 1  
    sum += i;
```

- 1~10에서 짝수만 더하는 예 (1)

```
int sum = 0, i;  
  
for(i=2; i<=10; i+=2)          // 초깃값이 2  
    sum += i;
```

홀수만 더할 때와 짝수만 더할 때
모두 i값을 2씩 증가시켰으나
초기값은 각각 다르게 해야 함.
이런 불편함을 해소하려면
continue문을 이용할 것

continue문

- 1~10에서 홀수만 더하는 예 (2) – continue문 이용

```
int sum = 0, i;

for(i=1; i<=10; i++)
{
    if(i%2 == 0) continue;    // 짝수면 아래 문장을 진행하지 않고 증감식으로 이동한다.
    sum += i;
}
```

- 1~10에서 짝수만 더하는 예 (2) – continue문 이용

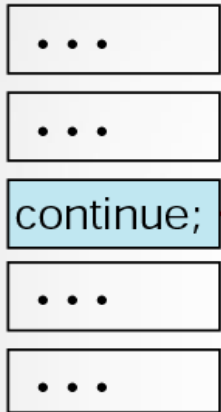
```
int sum = 0, i;

for(i=1; i<=10; i++)
{
    if(i%2 != 0) continue;    // 홀수면 아래 문장을 진행하지 않고 증감식으로 이동한다.
    sum += i;
}
```


continue문

while (limit test)

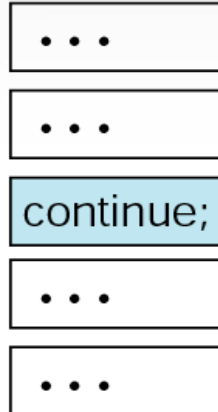
{



} // while

do

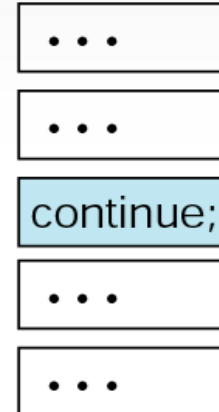
{



} while (limit test);

for (initialization; limit test; update)

{

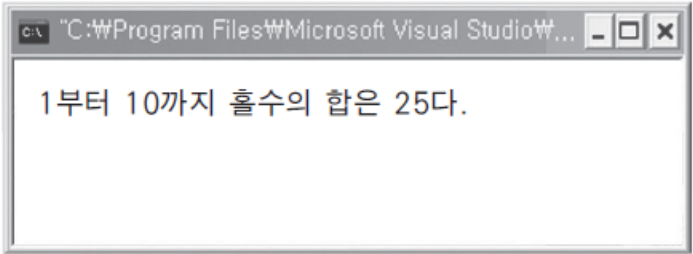


} // for

continue문

예제) continue문을 이용해서 홀수만 더하는 예제

```
01 #include <stdio.h>
02
03 int main(void)
04 {
05     int i, sum = 0;
06
07     for(i = 1; i <= 10; i++)
08     {
09         if(i % 2 == 0) continue;
10         sum += i;
11     }
12
13     printf("1부터 10까지 홀수의 합은 %d다.\n", sum);
14     return 0;
15 }
```



1부터 10까지 홀수의 합은 25다.

continue문

- 조건 2개를 동시에 비교하는 경우에도 유용하게 사용 가능
- 1 ~ 20에서 2의 배수이면서 3의 배수인 수를 제외하고 출력하는 프로그램

- 2의 배수인 조건 `(i % 2) == 0`

- 3의 배수인 조건 `(i % 3) == 0`

- 두 조건의 AND 연산 : 2의 배수이면서 3의 배수인 조건

`(i % 2) == 0 && (i % 3) == 0`

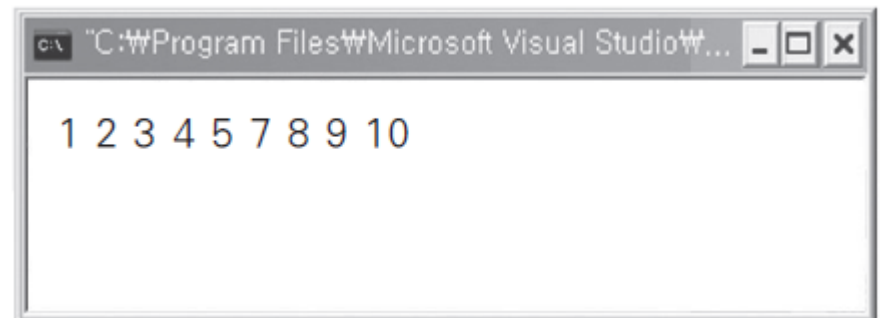
- 위의 조건을 만족하지 않을 때 값의 출력

```
for(i=1; i<=20; i++) {  
    if((i % 2 == 0) && (i % 3 == 0)) continue;  
    printf("%d", i);  
}
```

continue문

예제) 2의 배수이면서 3의 배수인 수를 제외하고 출력

```
01 #include <stdio.h>
02
03 int main(void)
04 {
05     int i;
06     for(i = 1; i <= 10; i++)
07     {
08         if((i % 2 == 0) && (i % 3 == 0)) continue;
09         printf("%d ", i);
10     }
11
12     printf("\n");
13     return 0;
14 }
15
```



```
C:\Program Files\Microsoft Visual Studio\...
1 2 3 4 5 7 8 9 10
```