

5장. 제어문

박 종 혁 교수

서울과학기술대학교 컴퓨터공학과

UCS Lab

Tel: 970-6702

Email: jhpark1@seoultech.ac.kr

목차

❖ 조건문

- if
- switch

❖ 반복문

- 반복문의 필요성
- for
- while
- do while
- 무한 루프

❖ 분기문

- break
- continue
- goto
- return

제어문

- 프로그램의 수행 순서를 제어하기 위한 목적의 문장

제어문의 종류	C 구문	설명
조건문	if	조건식이 참이면 문장을 수행한다.
	switch	정수식의 값에 따라 수행할 문장을 선택한다.
반복문	for	조건식이 참인 동안 문장을 반복 수행한다.
	while	
	do while	
분기문	break	switch나 반복문을 빠져나간다.
	continue	반복문의 시작으로 돌아간다.
	goto	지정된 레이블의 문장으로 이동한다.
	return	함수를 호출한 곳으로 돌아간다.

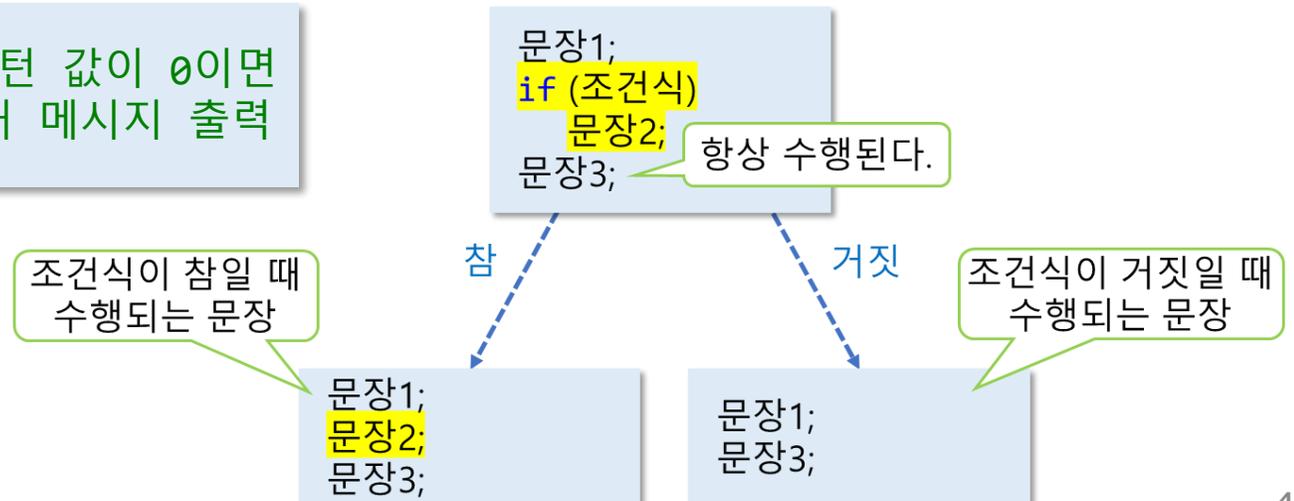
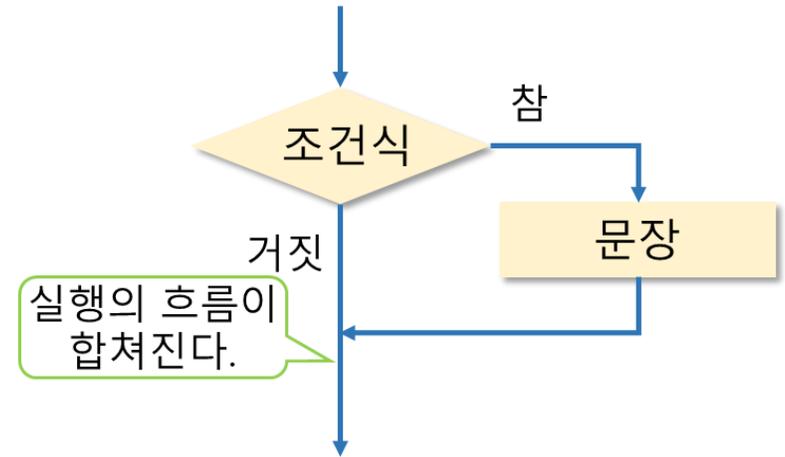
if문

형식 if (조건식)
문장;

사용예 if (num < 0)
printf("음수입니다.");

```
int price = 35000;  
if (month >= 6 && month <= 8) // 성수기(6~8월)이면  
    price = 70000;
```

```
if (! scanf("%d", &num)) // scanf 함수의 리턴 값이 0이면  
    printf("입력 실패"); // 입력 실패 시 에러 메시지 출력
```



예제 : if의 사용 예

```
03 int main(void)
04 {
05     int score;
06
07     printf("퀴즈 점수를 입력하세요: ");
08     scanf("%d", &score);
09
10     if (score < 70)
11         printf("재시!!!\n"); // score < 70인 경우에 수행된다.
12
13     // if문 다음 문장에서 실행의 흐름이 다시 만난다.
14     printf("다음 수업은 일주일 후입니다.\n");
15
16     return 0;
17 }
```

실행결과

퀴즈 점수를 입력하세요: 55
재시!!!
다음 수업은 일주일 후입니다.

재시 여부에 관계없이
항상 출력한다.

복합문

- { }으로 묶인 문장, 블록
- 언제든지 단일문 대신 사용할 수 있다

```
if (score < 70)    // 수행할 문장이 2개 이상이면 { }로 묶어준다.
{
    printf("재시!!!\n");
    printf("재시는 90점 이상이어야 통과입니다\n");
}
```

```
if (num < 0)
{
    printf("음수입니다.");
    num = -num;
}
```

```
if (num < 0) {
    printf("음수입니다.");
    num = -num;
}
```

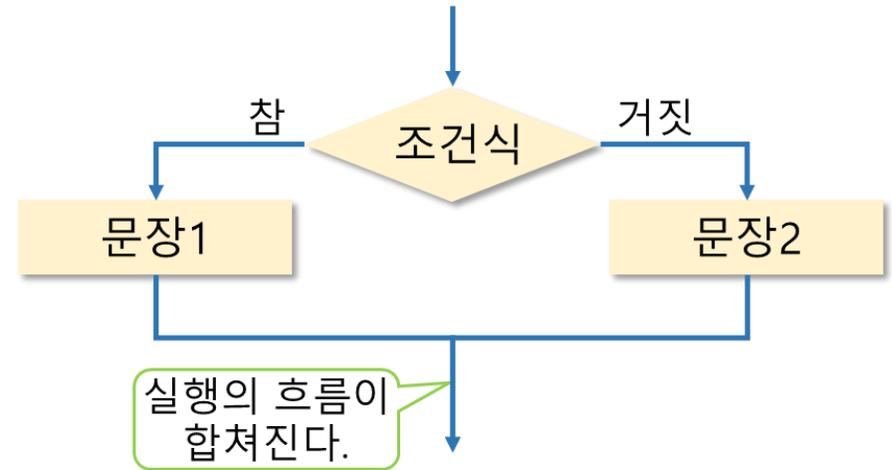
if else문

형식

```
if (조건식)
    문장1;
else
    문장2;
```

사용예

```
if (num < 0)
    printf("음수입니다.");
else
    printf("양수입니다.");
```



score < 70이 거짓인 경우에 해당

```
if (score < 70)
    printf("재시!!!\n");
if (score >= 70)
    printf("통과!!!\n");
```

```
if (score < 70)
    printf("재시!!!\n");
else // 그렇지 않으면
    printf("통과!!!\n");
```

조건 연산자

```
int max = (x > y) ? x : y;
```

```
int max;
if (x > y) // x가 y보다 크면
    max = x;
else // 그렇지 않으면
    max = y;
```

예제 : if else의 사용 예

```
03 int main(void)
04 {
05     int score;
06
07     printf("퀴즈 점수를 입력하세요: ");
08     scanf("%d", &score);
09
10     if (score < 70)
11         printf("재시!!!\n"); // score < 70인 경우에 수행된다.
12     else
13         printf("통과!!!\n"); // score >= 70인 경우에 수행된다.
14
15     return 0;
16 }
```

실행결과

퀴즈 점수를 입력하세요: 90
통과!!!

예제 : 0으로 나누기 검사

```
03  int main(void)
04  {
05      int x, y;
06
07      printf("정수 2개를 입력하세요: ");
08      scanf("%d %d", &x, &y);
09
10      if (y != 0)
11          printf("%d / %d = %f\n", x, y, (double)x / y);
12      else
13          printf("0으로 나눌 수 없습니다.\n");
14
15      return 0;
16  }
```

소수점 이하까지 몫을 구하려면
double로 형 변환한다.

실행결과

정수 2개를 입력하세요: 72 15
72 / 15 = 4.800000

중첩된 if

- if문 안에 포함된 if문

8세 이상이면 유료,
8세 미만은 무료

```
int fee;  
if (age >= 8)  
{  
    if (age >= 65  
        fee = 5000;  
    else  
        fee = 10000;  
}  
else  
{  
    fee = 0;  
}
```

65세 이상은 경로 우대,
아니면 정상 요금

// 중첩된 if

예제 : 중첩된 if의 사용 예

```
05     int age, fee;
06
07     printf("나이? ");
08     scanf("%d", &age);
09
10     if (age >= 8) {
11         if (age >= 65) { // if문 안에 다른 if문을 포함할 수 있다.
12             fee = 5000;
13         }
14         else {
15             fee = 10000;
16         }
17     }
18     else {
19         fee = 0;
20     }
21     printf("입장료: %d원\n", fee);
```

중첩된 if

실행결과

나이? 20

입장료: 10000원

else if문 (1/2)

- else의 블록 안에 다른 문장 없이 또 다른 if문만 들어 있을 때
- 여러 가지 조건을 순서대로 검사

```
if (age < 8) {  
    fee = 0;  
}  
else {  
    if (age >= 65) {  
        fee = 5000;  
    }  
    else {  
        fee = 10000;  
    }  
}
```

else 안에 또
다른 if문만
포함되어 있는
경우



```
if (age < 8) {  
    fee = 0;  
}  
else if (age >= 65) {  
    fee = 5000;  
}  
else {  
    fee = 10000;  
}
```

여러 가지 조건을
순서대로 검사한다

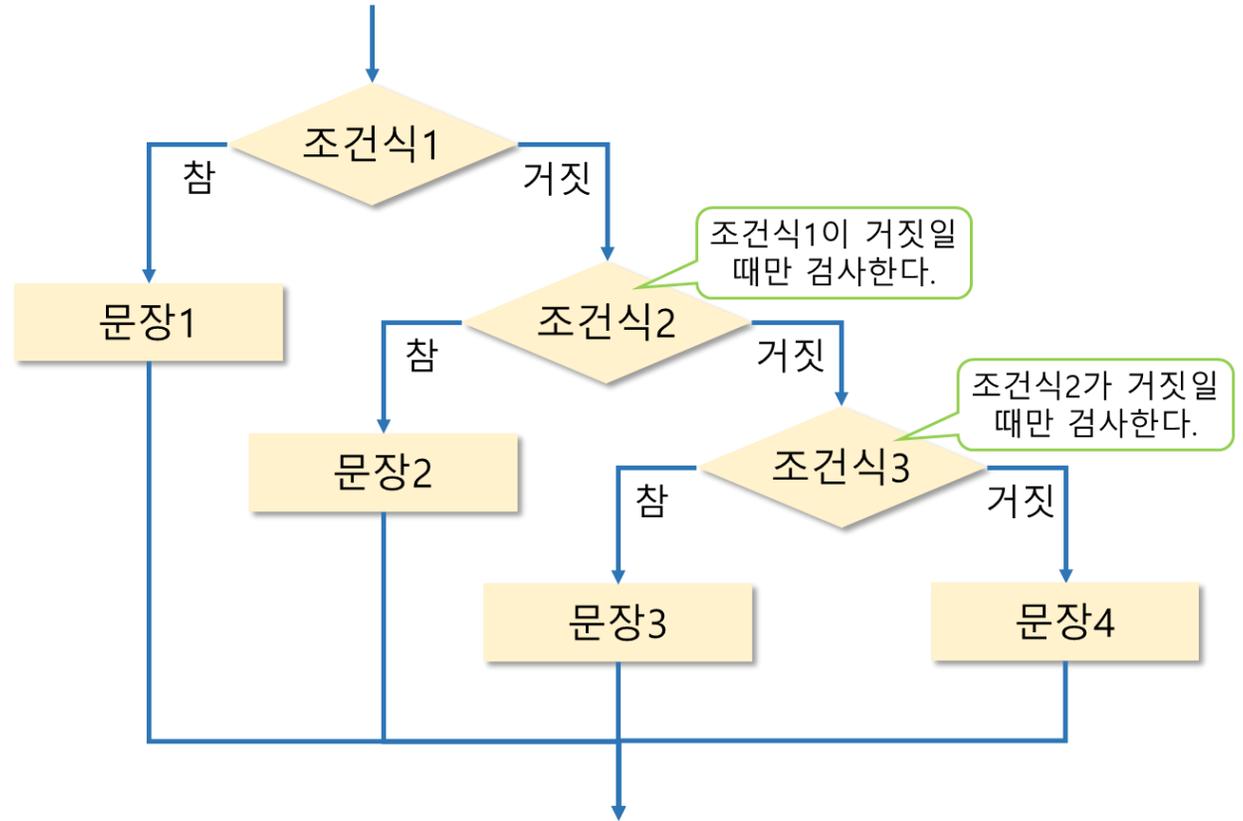
else if문 (2/2)

형식

```
if (조건식1)
    문장1;
else if (조건식2)
    문장2;
else
    문장3;
```

사용예

```
if (num < 0)
    printf("음수입니다.");
else if (num > 0)
    printf("양수입니다.");
else
    printf("0입니다.");
```



예제 : else if의 사용 예

```
05     int age, fee;
06
07     printf("나이? ");
08     scanf("%d", &age);
09
10     if (age < 8) {
11         fee = 0;
12     }
13     else if (age >= 65) {    // age >= 8 && age >= 65라는 의미
14         fee = 5000;
15     }
16     else {
17         fee = 10000;
18     }
19     printf("입장료: %d원\n", fee);
```

실행결과

나이? 20

입장료: 10000원

다중 if

- 서로 독립적인 조건을 여러 개 비교하는 경우
- 각각의 if문은 else if로 연결되지 않는다

```
fee = 10000;
```

```
if (age >= 65)  
    fee -= 5000;
```

65세 이상이면
5000원 할인

```
if (is_local == 1)  
    fee -= 1000;
```

지역 주민이면
1000원 할인

65세 이상과 지역 주민은
서로 독립적인 조건

65세 이상이면서
지역주민일 수도 있고
아닐 수도 있다.

if문의 비교

중첩된 if

```
if (age >= 8)
{
    if (age >= 65)
        fee = 5000;
    else
        fee = 10000;
}
else
{
    fee = 0;
}
```

age >= 8이
참일 때만 검사

else if

```
if (age < 8)
{
    fee = 0;
}
else if (age >= 65)
{
    fee = 5000;
}
else
{
    fee = 10000;
}
```

age < 8이 거짓일 때만 검사

다중 if

```
fee = 10000;
if (age >= 65)
{
    fee -= 5000;
}
if (is_local == 1)
{
    fee -= 1000;
}
```

age >= 65의 참,
거짓 여부에 관계 없이 검사

else if의 활용

- ❖ else if문은 선택적으로 코드를 수행하게 만들 때 유용하게 사용된다
 - 텍스트 기반의 메뉴 처리

예제 : else if를 이용한 텍스트 기반의 메뉴 처리

```
03 int main(void)
04 {
05     int menu;                // 선택된 메뉴 번호
06     char filename[32] = "test.avi"; // 디폴트로 재생할 파일 이름
07
08     printf("1.파일 열기\n");    // 메뉴를 출력한다.
09     printf("2.재생\n");
10     printf("3.재생 옵션\n");
11     printf("선택: ");
12
13     scanf("%d", &menu);        // 메뉴 번호를 입력받는다.
14     if (menu == 1) {          // 파일 열기 메뉴
15         printf("재생할 파일 이름? ");
16         scanf("%s", filename);
17     }
18     else if (menu == 2) {     // 재생 메뉴
19         printf("%s를 재생합니다.\n", filename);
20     }
21     else if (menu == 3) {    // 재생 옵션 메뉴
22         printf("재생 옵션을 선택합니다.\n");
23     }
24     else {                   // 1~3이 아닌 메뉴 번호를 선택하는 경우
25         printf("잘못 선택하셨습니다.\n");
26     }
27
28     return 0;
29 }
```

실행결과

1.파일 열기

2.재생

3.재생 옵션

선택: 1

재생할 파일 이름? movie.avi

예제 : else if를 이용한 사칙연산 계산기

```
03 int main(void)
04 {
05     int a, b;        // 피연산자
06     char op;        // 연산자 기호를 문자로 저장할 변수
07
08     printf("수식? ");
09     scanf("%d %c %d", &a, &op, &b);    // 10 + 30 형태로 입력받는다.
10
11     if (op == '+') {
12         printf("%d + %d = %d\n", a, b, a + b);
13     }
14     else if (op == '-') {
15         printf("%d - %d = %d\n", a, b, a - b);
16     }
17     else if (op == '*') {
18         printf("%d * %d = %d\n", a, b, a * b);
19     }
20     else if (op == '/') {
21         if (b != 0) // 중첩된 if
22             printf("%d / %d = %.2f\n", a, b, (double)a / b);
23         printf("0으로 나눌 수 없습니다.\n");
24     }
25     else {          // +, -, *, /가 아닌 경우
26         printf("잘못된 수식입니다.\n");
27     }
28
29     return 0;
30 }
```

실행결과

수식? 78 + 21
78 + 21 = 99

피연산자와 연산자 사이에
빈칸을 넣어도 되고 안 넣어도 된다.

a/b를 실수로 구하려면
double로 형 변환

switch문

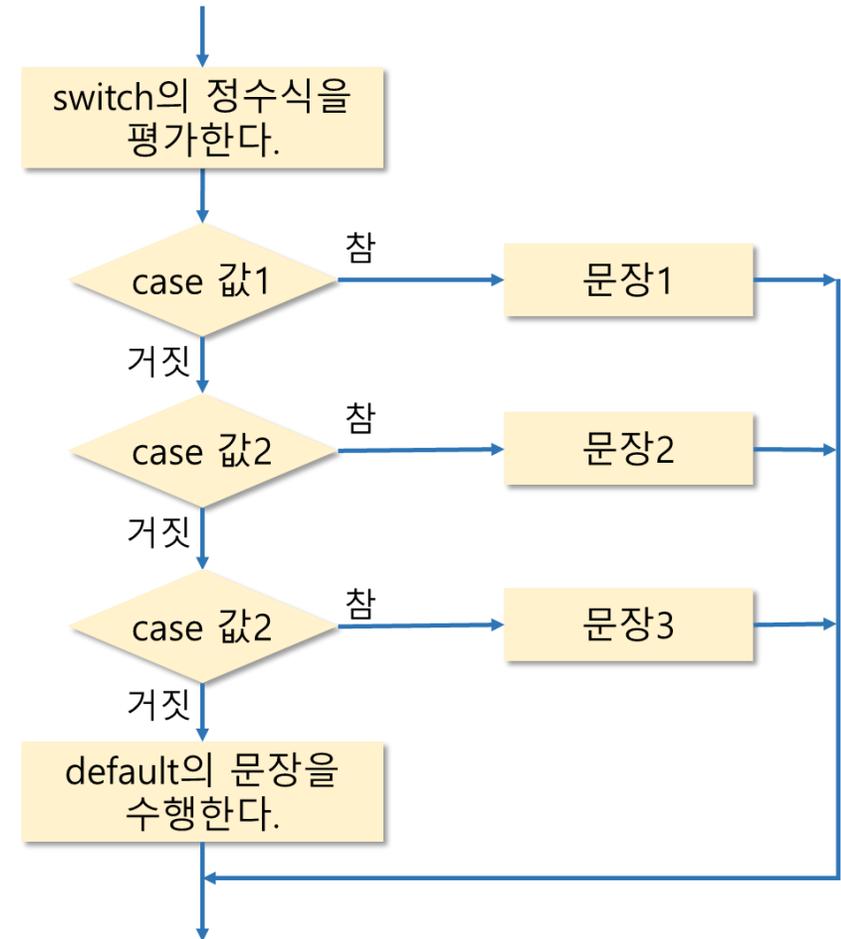
- 정수식의 값에 따라서 여러 가지 경우 중 하나로 분기한다

형식

```
switch (정수식) {  
  case 정수값1 :  
    문장1;  
    break;  
  case 정수값2 :  
    문장2;  
    break;  
  ⋮  
  default :  
    문장n;  
    break;  
}
```

사용예

```
switch (menu) {  
  case 1:  
    printf("1번 메뉴 선택");  
    break;  
  case 2:  
    printf("2번 메뉴 선택");  
    break;  
  default:  
    printf("잘못 선택하셨습니다.\n");  
    break;  
}
```



예제 : switch를 이용한 텍스트 기반의 메뉴 처리

```
03  int main(void)
04  {
05      int menu;                // 선택된 메뉴 번호
06      char filename[32] = "test.avi"; // 디폴트로 재생할 파일 이름
07
08      printf("1.파일 열기\n");    // 메뉴를 출력한다.
09      printf("2.재생\n");
10      printf("3.재생 옵션\n");
11      printf("선택: ");
12
13      scanf("%d", &menu);        // 메뉴 번호를 입력받는다.
14      switch (menu) {
15      case 1:
16          printf("재생할 파일 이름? ");
17          scanf("%s", filename);
18          break;
19      case 2:
20          printf("%s를 재생합니다.\n", filename);
21          break;
22      case 3:
23          printf("재생 옵션을 선택합니다.\n");
24          break;
25      default:                    // 1~30이외의 메뉴 번호 선택 시
26          printf("잘못 선택하셨습니다.\n");
27          break;
28      }
29
30      return 0;
31  }
```

실행결과

- 1.파일 열기
- 2.재생
- 3.재생 옵션

선택: 1

재생할 파일 이름? movie.avi

switch의 수행 순서

menu에 2가 입력된 경우 switch의 수행 순서

```
switch (menu) {  
  case 1:  
    printf("재생할 파일 이름? ");  
    scanf("%s", filename);  
    break;  
  case 2:  
    printf("%s를 재생합니다.\n", filename);  
    break;  
  case 3:  
    printf("선택을 선택합니다.\n");  
    break;  
  default:  
    printf("잘못 선택하셨습니다.\n");  
    break;  
}
```

① 정수식의 값을 평가한다

② 값이 일치하는 case를 찾는다

③ case 다음 문장을 수행한다

④ switch를 탈출한다

예제 : switch를 이용한 사칙연산 계산기

```
03 int main(void)
04 {
05     int a, b;           // 피연산자
06     char op;           // 연산자 기호를 문자로 저장할 변수
07
08     printf("수식? ");
09     scanf("%d %c %d", &a, &op, &b); // 10 + 30 형태로 입력받는다.
10
11     switch (op) {
12     case '+':
13         printf("%d + %d = %d\n", a, b, a + b);
14         break;
15     case '-':
16         printf("%d - %d = %d\n", a, b, a - b);
17         break;
18     case '*':
19         printf("%d * %d = %d\n", a, b, a * b);
20         break;
21     case '/':
22         if (b != 0)
23             printf("%d / %d = %.2f\n", a, b, (double)a / b);
24         else
25             printf("0으로 나눌 수 없습니다.\n");
26         break;
27     default: // +, -, *, /가 아닌 경우
28         printf("잘못된 수식입니다.\n");
29         break;
30     }
31
32     return 0;
33 }
```

실행결과

수식? 78 + 21

78 + 21 = 99

else if vs. switch (1/2)

비교할 값이 2개 미만일 때는 if를 사용한다

```
switch (num % 2) {  
  case 0:  
    printf("even\n");  
    break;  
  case 1:  
    printf("odd\n");  
    break;  
}
```

```
if (num % 2 == 0) {  
  printf("even\n");  
}  
else {  
  printf("odd\n");  
}
```

==로 비교할 값이
2개 미만인 경우

비교할 값이 2개 이상일 때는 switch를 사용한다

```
if (menu == 1) {  
  printf("메뉴1");  
}  
else if (menu == 2) {  
  printf("메뉴2");  
}  
else if (menu == 3) {  
  printf("메뉴3");  
}  
else {  
  printf("에러");  
}
```

```
switch (menu) {  
  case 1:  
    printf("메뉴1");  
    break;  
  case 2:  
    printf("메뉴2");  
    break;  
  case 3:  
    printf("메뉴3");  
    break;  
  default:  
    printf("에러");  
    break;  
}
```

==로 비교할 값이 2
개 이상인 경우

else if vs. switch (2/2)

범위를 비교할 때는 else if를 사용한다.

```
switch (score/10) {  
  case 9:  
    grade = 'A';  
    break;  
  case 8:  
    grade = 'B';  
    break;  
  case 7:  
    grade = 'C';  
    break;  
  case 6:  
    grade = 'D';  
    break;  
  default:  
    grade = 'F';  
    break;  
}
```

```
if (score >= 90) {  
  grade = 'A';  
}  
else if (score >= 80) {  
  grade = 'B';  
}  
else if (score >= 70) {  
  grade = 'C';  
}  
else if (score >= 60) {  
  grade = 'D';  
}  
else {  
  grade = 'F';  
}
```

단순한 정수 값
비교가 아닌 경우

switch 사용 시 주의 사항 (1/3)

❖ switch문에서 break는 생략할 수 있다

- 실수로 break를 빠뜨리면, break를 만나거나 switch의 끝을 만날 때까지 나타나는 모든 문장들을 수행한다

```
switch (day_of_week) {           // 항상 0~6사이의 값
case 1:                          // mon
    fee = 5000;                  // 월요일은 할인 요금
    break;
case 6:                          // sat
case 0:                          // sun
    fee = 10000;                 // 토,일은 주말 요금
    break;
default:
    fee = 8000;                  // 화~목은 평일 요금
    break;
}
```

day_of_week가 6일 때와 0일 때 같은 코드를 수행한다.

의도적으로 특정 case에 대한 break를 생략할 수도 있다

switch 사용 시 주의 사항 (2/3)

❖ default도 생략할 수 있다

- switch문에 일치하는 case가 없고 default도 없으면, 아무것도 수행하지 않고 switch문을 빠져나간다

❖ default는 switch문의 맨 마지막에 작성하며, default에도 break를 써주는 것이 좋다

menu에 5가 입력된 경우 switch의 수행 순서

① 정수식의 값을 평가한다

③ default 다음 문장을 수행한다

② 값이 일치하는 case를 찾기 못하면 default를 찾는다

④ switch를 탈출한다

```
switch (menu) {
default:
    printf("잘못 선택하셨습니다.\n");
    break;
case 1:
    printf("재생할 파일 이름? ");
    scanf("%s", filename);
    break;
case 2:
    printf("%s를 재생합니다.\n", filename);
    break;
case 3:
    printf("재생 옵션을 선택합니다.\n");
    break;
}
```

} default는 switch문의 어디에도 올 수 있다

switch 사용 시 주의 사항 (3/3)

- switch의 () 안에는 정수식만 사용할 수 있으며, 실수나 문자열은 사용할 수 없다

```
double x;  
switch (x) {           // 실수는 switch에서 사용할 수 없다.  
case 0.1:             // case문에 정수 값만 사용할 수 있다.  
    ...  
    break;  
}
```

- 특정 case에서만 사용되는 변수를 선언하려면 { }를 쓰고 그 안에 변수 선언 및 필요한 문장을 적어준다

```
switch (menu) {  
case 2:  
    {  
        int seconds;           // case 2에서만 사용되는 변수 선언  
        scanf("%d", &seconds);  
        printf("%s를 %d초간 재생합니다.\n", filename, seconds);  
        break;  
    }  
    ...  
}
```

case 안에서 사용될 변수를 선언하려면 { }가 필요하다.

반복문

- 조건이 만족하는 동안 주어진 문장을 반복해서 수행
→ 루프(loop)
- 반복문을 이용하면 사용자가 원하는 만큼 반복적으로 수행

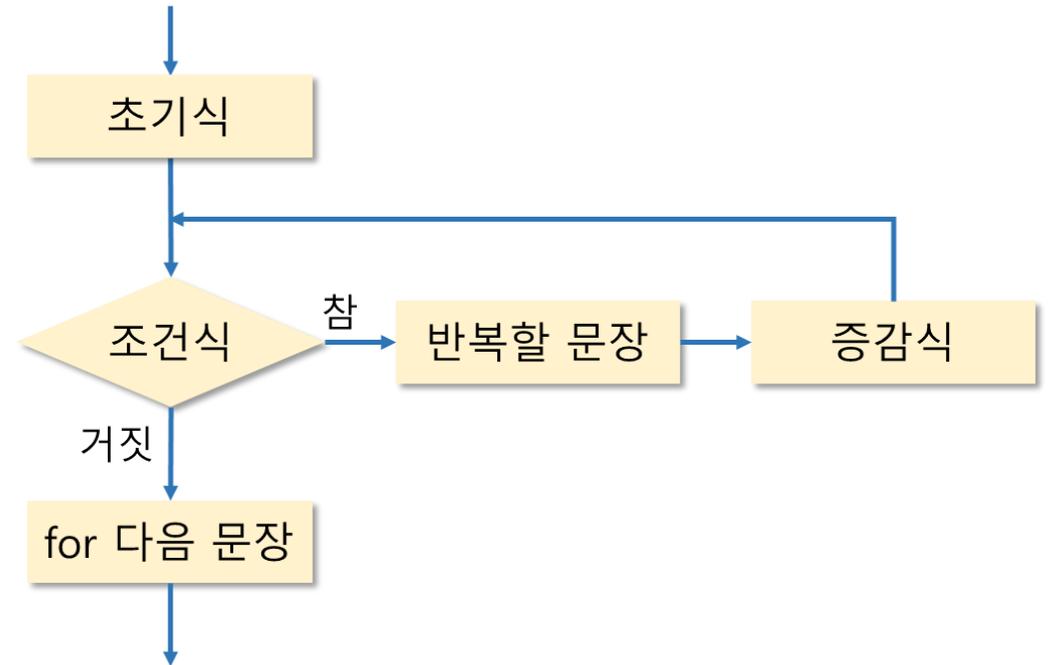
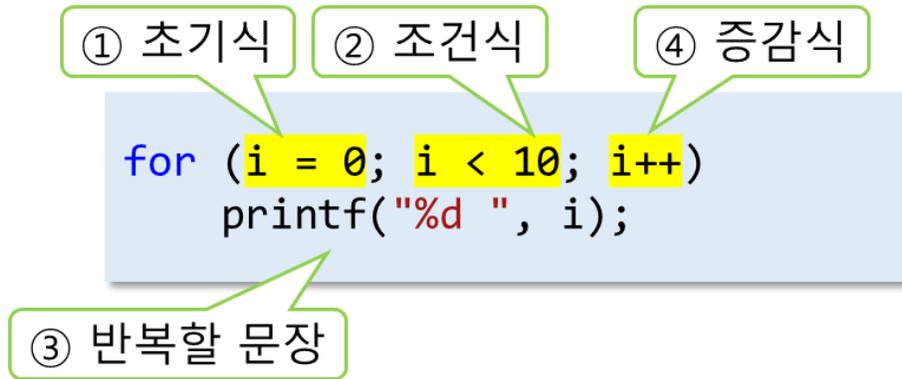
```
while (!종료조건) { // 종료조건이 거짓인 동안 반복 수행한다.  
    사칙연산 계산을 수행한다.  
}
```

- for
- while
- do while

for문

형식 **for** (초기식; 조건식; 증감식)
 반복할문장;

사용예 `for (i = 0; i < 10; i++)`
 `printf("%d ", i);`



수행 순서: ① → ② → ③ → ④ → ② → ③ → ④ → ... → ②

예제 : for의 사용 예

```
01  #include <stdio.h>
02
03  int main(void)
04  {
05      int i;
06      for (i = 0; i < 10; i++) // i가 10이 되면 루프 탈출
07          printf("%d ", i);    //반복할 문장
08      printf("\n");
09
10      return 0;
11  }
```

초기식 조건식 증감식

실행결과

0 1 2 3 4 5 6 7 8 9

for문의 루프 제어 변수

- 일반적으로 어떤 문장을 N번 반복 수행하는 용도로 사용된다

```
for (i = 0; i < 10; i++)  
    printf("%d ", i);
```

10번 반복한다

- for문의 반복 회차마다 루프 제어 변수의 값이 변경되어, 특정 시점에 for의 조건식이 참이 되어 루프를 탈출할 수 있으면 된다

```
for (i = 10; i > 0; --i)           // 루프 제어 변수를 10에서부터 1씩 감소시킨다.  
    printf("%d ", i);
```

```
for (i = 0; i < 20; i += 2)       // 루프 제어 변수를 0에서부터 2씩 증가시킨다.  
    printf("%d ", i);
```

```
for (j = 1; j <= 1000000; j *= 10) // 루프 제어 변수를 1에서부터 10씩 곱한다.  
    printf("%7d\n", j);
```

예제 : 입력된 정수들의 합계 구하기

```
03  int main(void)
04  {
05      int num = 0;    // 입력받은 정수를 저장할 변수
06      int sum = 0;    // 합계를 저장할 변수
07      int i;         // 루프 제어 변수
08
09      printf("정수 5개를 입력하세요: ");
10      for (i = 0; i < 5; i++) // i가 5가 되면 루프 탈출
11      {
12          scanf("%d", &num); // i가 0, 1, 2, 3, 4일 때 반복 수행
13          sum += num;
14      }
15      printf("합계: %d\n", sum);
16
17      return 0;
18  }
```

sum에 합계를 구해야 하므로 0으로 초기화한다.

실행결과

```
정수 5개를 입력하세요: 12 32 45 63 7
합계: 159
```

for문의 여러 가지 변형

- for문에서 루프 제어 변수를 여러 개 사용할 수도 있다

```
for (i = 0, j = 100; i < 10 && j > 0; i++, j /= 10)
    printf("i = %d, j = %d\n", i, j);
```

- 초기식, 조건식, 증감식과 반복할 문장은 모두 생략할 수 있다

for의 초기식을 생략할 수 있다.

```
for (; i < 10; i++)
    printf("i = %d ", i);
```

for의 조건식을 생략할 수 있다.

```
for (i = 0; ; i++)
    printf("i = %d ", i);
```

```
for (i = 0; i < 10; )
```

```
;
```

for의 반복할 문장을 생략할 수 있다.

for의 증감식을 생략할 수 있다.

```
for (i = 0; i < 10; )
    printf("i = %d ", i++);
```

for의 초기식, 조건식, 증감식을 모두 생략할 수 있다.

```
for ( ; ; )
    printf("i = %d ", i);
```

무한 루프

널 문장

- 처리할 내용 없이 세미콜론(;)만으로 이루어진 문장

```
if (a > 0);           // 실수로 세미콜론을 쓰는 경우
    a = -a;           // if문의 다음 문장이 된다.
```

- for문의 초기식, 조건식, 증감식에서 필요한 기능을 모두 수행해서 반복할 문장을 따로 지정할 필요가 없을 경우에 유용

```
for (i = 0; i < 10; printf("%d\n", i++)) // 증감식으로 printf 함수 호출문이 사용된 경우
;           // 다음 줄에 세미콜론을 적어서 널 문장을 명확하게 표시하는 것이 좋다.
```

```
for (i = 0; i < 10; printf("%d\n", i++)) {}
// 빈 블록을 만들어서 수행할 문장이 없다고 명시적으로 표시할 수 있다.
```

예제 : 입력된 문자로 직사각형 그리기

```
03 int main(void)
04 {
05     int width, height;
06     char ch;
07     int i, j;
08
09     printf("직사각형의 폭과 높이? ");
10     scanf("%d %d", &width, &height);
11     printf("직사각형을 그릴 문자? ");
12     scanf("%c", &ch); // %c앞에 빈칸 지정
13
14     for (i = 0; i < height; i++)
15     {
16         for (j = 0; j < width; j++) // 중첩된 for
17             printf("%c", ch);
18         printf("\n");
19     }
20
21     return 0;
22 }
```

입력 버퍼에 남아 있는 공백 문자(\n, ', \t)를 무시한다.

실행결과

직사각형의 폭과 높이? 20 3

직사각형을 그릴 문자? *

#HW 2: For문을 이용 피라미드를 화면에 출력하기

- 기한: 1주 (~4/7)

- 입력:

 - 층수 입력 (5)

- 출력예시

*

**

while문

형식

while (조건식)
반복할문장;

사용예

```
i = 0;  
while (i < 10)  
    printf("%d ", i++);
```

① 초기식

② 조건식

④ 증감식

```
for (i = 0; i < 10; i++)  
    printf("%d ", i);
```

③ 반복할 문장

for의 수행 순서

① → ② → ③ → ④ → ② → ③ → ④ → ... → ②

반복할 문장

조건식

참

거짓

do while 다음 문장

① 조건식

```
i = 0;  
while (i < 10)  
    printf("%d ", i++);
```

② 반복할 문장

while의 수행 순서

① → ② → ① → ② → ... → ①

예제 : while의 사용 예

```
01  #include <stdio.h>
02
03  int main(void)
04  {
05      int i = 0;
06      while (i < 10)           // i가 10이 되면 루프 탈출
07          printf("%d ", i++); //반복할 문장
08      printf("\n");
09
10      return 0;
11  }
```

실행결과

0 1 2 3 4 5 6 7 8 9

예제 : 입력된 정수들의 합계 구하기

```
03 int main(void)
04 {
05     int num = 0;
06     int sum = 0;
07     int i = 0; // for의 초기식
08
09     printf("정수 5개를 입력하세요: ");
```

```
10     while (i < 5) // 조건식
11     {
12         // 반복할 문장
13         scanf("%d", &num);
14         sum += num;
15         i++; // for의 증감식
16     }
17     printf("합계: %d\n", sum);
18
19     return 0;
20 }
```

실행결과

```
정수 5개를 입력하세요: 12 32 45 63 7
합계: 159
```

예제 : 반복 수행되는 사칙연산 계산기

```
03 int main(void)
04 {
05     int a, b;
06     char op;
07     char yesno = 'Y'; // 계속 수행할지를 나타내는 변수
08
09     while (yesno == 'Y' || yesno == 'y')
10     {
11         printf("수식? ");
12         scanf("%d %c %d", &a, &op, &b); // 10 + 30 형태로 입력받는다.
13
14         switch (op) {
15             case '+':
16                 printf("%d + %d = %d\n", a, b, a + b);
17                 break;
18             case '-':
19                 printf("%d - %d = %d\n", a, b, a - b);
20                 break;
21             case '*':
22                 printf("%d * %d = %d\n", a, b, a * b);
23                 break;
24             case '/':
25                 if (b != 0)
26                     printf("%d / %d = %.2f\n", a, b, (double)a / b);
27                 else
28                     printf("0으로 나눌 수 없습니다.\n");
29                 break;
30             default: // +, -, *, /가 아닌 경우
31                 printf("잘못된 수식입니다.\n");
32                 break;
33         }
34         printf("계속 하시겠습니까(Y/N)? ");
35         scanf("%c", &yesno);
36     }
37     return 0;
38 }
39 }
```

처음에 while문을 수행해야 하므로 'Y'로 초기화 한다.

공백 문자를 무시하고 문자를 입력받으려면 %c앞에 빈칸을 지정한다.

실행결과

수식? 3 + 10

3 + 10 = 13

계속 하시겠습니까(Y/N)? y

수식? 78 * 5

78 * 5 = 390

계속 하시겠습니까(Y/N)? y

수식? 56 & 3

잘못된 수식입니다.

계속 하시겠습니까(Y/N)? n

do while문

형식

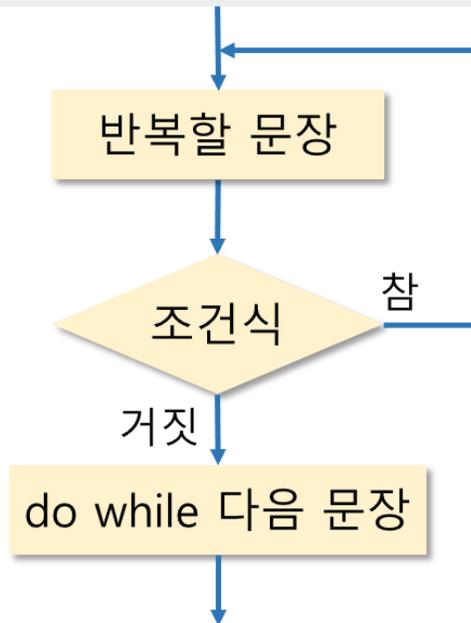
do

반복할문장;

while (조건식);

사용예

```
i = 0;  
do  
    printf("%d ", i++);  
while (i < 10);
```



```
i = 0;  
do  
    printf("%d ", i++);  
while (i < 10)
```

① 반복할 문장

② 조건식

do while의 수행 순서

①→②→①→②→...→②

예제 : do while의 사용 예

```
01  #include <stdio.h>
02
03  int main(void)
04  {
05      int i = 0;
06      do
07          printf("%d ", i++); //반복할 문장
08      while (i < 10);         // i가 10이 되면 루프 탈출
09      printf("\n");
10
11      return 0;
12  }
```

실행결과

0 1 2 3 4 5 6 7 8 9

for, while, do while 비교

for	while	do while
<pre>for (i = 0; i < 0; i++) printf("%d ", i);</pre>	<pre>i = 0; while (i < 0) printf("%d ", i++);</pre>	<pre>i = 0; do printf("%d ", i++); while (i < 0);</pre>
실행 결과	실행 결과	실행 결과
[실행 결과 없음]	[실행 결과 없음]	0
문장이 한번도 수행되지 않는다.	문장이 한번도 수행되지 않는다.	반드시 한번은 수행된다.

- 반복문의 끝 부분에서 반복문의 탈출 여부를 결정해야 하는 프로그래머에서는 while보다는 do while을 사용하는 것이 자연스럽다

예제 : do while을 이용한 사칙연산 계산기

```
03 int main(void)
04 {
05     int a, b;
06     char op;
07     char yesno; // yesno의 초기화를 생략할 수 있다.
08
09     do { // 일단 한번 수행한다.
10         printf("수식? ");
11         scanf("%d %c %d", &a, &op, &b); // 10 + 30 형태로 입력받는다.
12
13         switch (op) {
14             case '+':
15                 printf("%d + %d = %d\n", a, b, a + b);
16                 break;
17             case '-':
18                 printf("%d - %d = %d\n", a, b, a - b);
19                 break;
20             case '*':
21                 printf("%d * %d = %d\n", a, b, a * b);
22                 break;
23             case '/':
24                 if (b != 0)
25                     printf("%d / %d = %.2f\n", a, b, (double)a / b);
26                 else
27                     printf("0으로 나눌 수 없습니다.\n");
28                 break;
29             default: // +, -, *, /가 아닌 경우
30                 printf("잘못된 수식입니다.\n");
31                 break;
32         }
33         printf("계속 하시겠습니까(Y/N)? ");
34         scanf(" %c", &yesno);
35     } while (yesno == 'Y' || yesno == 'y');
36
37     return 0;
38 }
```

실행결과

수식? 3 + 10

3 + 10 = 13

계속 하시겠습니까(Y/N)? y

수식? 78 * 5

78 * 5 = 390

계속 하시겠습니까(Y/N)? y

수식? 56 & 3

잘못된 수식입니다.

계속 하시겠습니까(Y/N)? n

루프 탈출 위치

while

```
while (루프탈출조건)
{
  문장1;
  문장2;
  문장3;
  :
}
```

루프 시작 부분에서
만 탈출할 수 있다.

do while

```
do {
  문장1;
  문장2;
  문장3;
  :
} while (루프탈출조건);
```

루프 끝 부분에서
만 탈출할 수 있다.

break를 함께 사용하는 경우

```
while (1)
{
  문장1;
  문장2;
  if(루프탈출조건)
    break;
  문장3;
  :
}
```

무한 루프

원하는 위치에서
언제든지 탈출할
수 있다.

무한 루프

❖ for에서는 무한 루프를 만들기 위해서 조건식을 생략한다

- for (; ;)

❖ while에서는 조건식을 생략할 수 없는 대신에 항상 참인 값을 조건식에 써준다

- while (1)

```
while (1)           // 무한 루프
{
    printf("infinite loop");
}
```

❖ 무한 루프를 수행중인 프로그램은 Ctrl+C로 강제 종료해야 한다

- 무한 루프를 안전하게 탈출하려면 break를 이용한다

예제 : 무한 루프를 이용한 메뉴 처리

```
03 int main(void)
04 {
05     int menu;
06     char filename[32] = "test.avi";
07
08     while (1) // 무한 루프이므로 Ctrl+C로 강제 종료해야 한다.
09     {
10         printf("1.파일 열기\n");
11         printf("2.재생\n");
12         printf("3.재생 옵션\n");
13         printf("선택: ");
14
15         scanf("%d", &menu);
16         switch (menu) {
17             case 1:
18                 printf("재생할 파일 이름? ");
19                 scanf("%s", filename);
20                 break;
21             case 2:
22                 printf("%s를 재생합니다.\n", filename);
23                 break;
24             case 3:
25                 printf("재생 옵션을 선택합니다.\n");
26                 break;
27             default:
28                 printf("잘못 선택하셨습니다.\n");
29                 break;
30         }
31     }
32
33     return 0;
34 }
```

실행결과

1.파일 열기

2.재생

3.재생 옵션

선택: 1

재생할 파일 이름? test.avi

1.파일 열기

2.재생

3.재생 옵션

선택:

Ctrl+C로
강제 종료해야 한다.

분기문

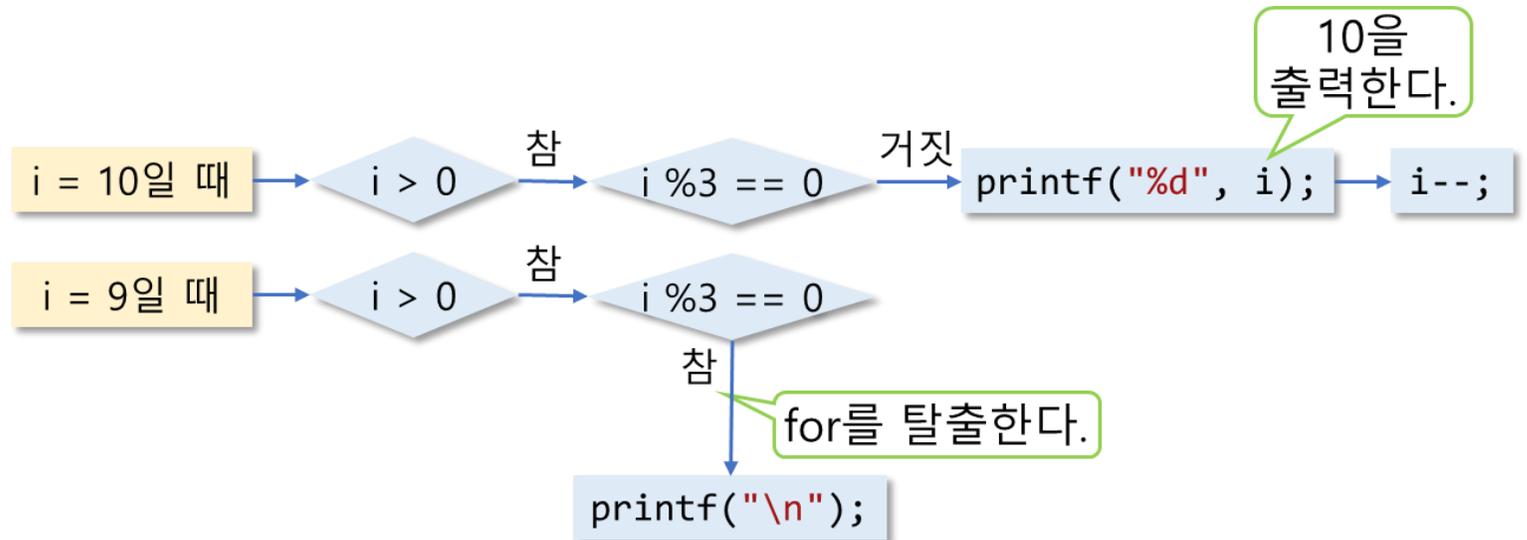
- 문장의 실행 순서를 변경
- break
- continue
- goto
- return

break문

- switch문 안에 사용하면 제어의 흐름이 switch를 탈출해서 switch의 다음 문장으로 이동한다
- for, while, do while 등의 반복문 안에서 사용하면 반복문을 빠져나가게 된다

```
for (i = 10; i > 0; i--)  
{  
    if (i % 3 == 0)  
        break;  
    printf("%d ", i);  
}  
printf("\n");
```

10



예제 : break의 사용 예

```
03 int main(void)
04 {
05     int i;
06
07     for (i = 10; i > 0; i--)
08     {
09         if (i % 3 == 0) // 루프 탈출 조건
10             break;
11         printf("%d ", i);
12     }
13     printf("\n");
14
15     return 0;
16 }
```

실행결과

10

break의 활용

- 무한 루프와 break를 이용하면 원하는 위치에서 루프 탈출 조건을 검사해서 루프를 빠져나올 수 있다

```
while (1) {  
    printf("1.파일 열기\n");  
    printf("2.재생\n");  
    printf("3.재생 옵션\n");  
    printf("선택: ");  
    scanf("%d", &menu);  
  
    if (menu == 0) break;  
  
    switch (menu) {  
    case 1:  
        printf("재생할 파일 이름? ");  
        scanf("%s", filename);  
        break;  
    :  
    }  
}
```

menu를 입력받은
다음 루프 탈출
조건을 검사한다.

예제 : 종료 메뉴를 가진 메뉴 처리 프로그램

```
03 int main(void)
04 {
05     int menu;
06     char filename[32] = "test.avi";
07
08     while (1)
09     {
10         printf("0.종료\n");
11         printf("1.파일 열기\n");
12         printf("2.재생\n");
13         printf("3.재생 옵션\n");
14         printf("선택: ");
15
16         scanf("%d", &menu);
17         if (menu == 0) // menu를 입력받은 다음 루프 탈출 조건을 검사한다.
18             break;
19         switch (menu) {
20             case 1:
21                 printf("재생할 파일 이름? ");
22                 scanf("%s", filename);
23                 break;
24             case 2:
25                 printf("%s를 재생합니다.\n", filename);
26                 break;
27             case 3:
28                 printf("재생 옵션을 선택합니다.\n");
29                 break;
30             default:
31                 printf("잘못 선택하셨습니다.\n");
32                 break;
33         }
34     }
35     printf("프로그램을 종료합니다.\n");
36 }
```

실행결과

1.파일 열기
2.재생
3.재생 옵션
선택: 1
재생할 파일 이름? test.avi
1.파일 열기
2.재생
3.재생 옵션
선택: 0
프로그램을 종료합니다.

종료 메뉴를 선택해서
프로그램을 종료한다.

센티널 값을 이용한 루프 탈출

❖ **센티널** : 입력되는 데이터의 끝을 나타내는 특별한 값

- 특정 값을 입력하면 프로그램을 종료하기로 미리 약속

```
while (1) {  
    printf("수식? ");  
    scanf("%d %c %d", &a, &op, &b);  
  
    if (a == 0 && op == '0' && b == 0)  
        break;  
  
    switch (op) {  
    case '+':  
        printf("%d + %d = %d\n", a, b, a + b);  
        break;  
    :  
    }  
}
```

"0 0 0"이 입력되면
무한 루프를 탈출한다.

예제 : 센티널 값을 이용한 사칙연산 계산기

```
08 while (1) 무한 루프
09 {
10     printf("수식? ");
11     scanf("%d %c %d", &a, &op, &b);
12
13     // 0 0 0이 입력되면 루프를 빠져나간다.
14     if (a == 0 && op == '0' && b == 0)
15         break; 루프 탈출 조건
16
17     switch (op) {
18     case '+':
19         printf("%d + %d = %d\n", a, b, a + b);
20         break;
21     case '-':
22         printf("%d - %d = %d\n", a, b, a - b);
23         break;
```

실행결과

수식? 3 + 10

3 + 10 = 13

수식? 78 * 5

78 * 5 = 390

수식? 56 & 3

잘못된 수식입니다.

수식? 0 0 0

센티널 값이 입력되면
종료한다.

continue문

- 루프의 시작이나 끝 부분으로 이동한다

for

루프 시작 부분
으로 이동한다.

```
for (i = 10; i > 0; i--)  
{  
    if (i % 3 == 0)  
        continue;  
    printf("%d ", i);  
}
```

while

루프 시작 부분
으로 이동한다.

```
i = 10;  
while (i > 0)  
{  
    if (i % 3 == 0)  
        continue;  
    printf("%d ", i);  
    i++;  
}
```

do while

루프 끝 부분
으로 이동한다.

```
i = 10;  
do  
{  
    if (i % 3 == 0)  
        continue;  
    printf("%d ", i);  
    i++;  
} while (i > 0);
```

예제 : continue의 사용 예

```
03  int main(void)
04  {
05      int i;
06
07      for (i = 10; i > 0; i--)
08      {
09          if (i % 3 == 0)
10              continue;           // 루프의 시작 부분으로 이동한다.
11          printf("%d ", i);
12      }
13      printf("\n");
14
15      return 0;
16  }
```

실행결과

10 8 7 5 4 2 1

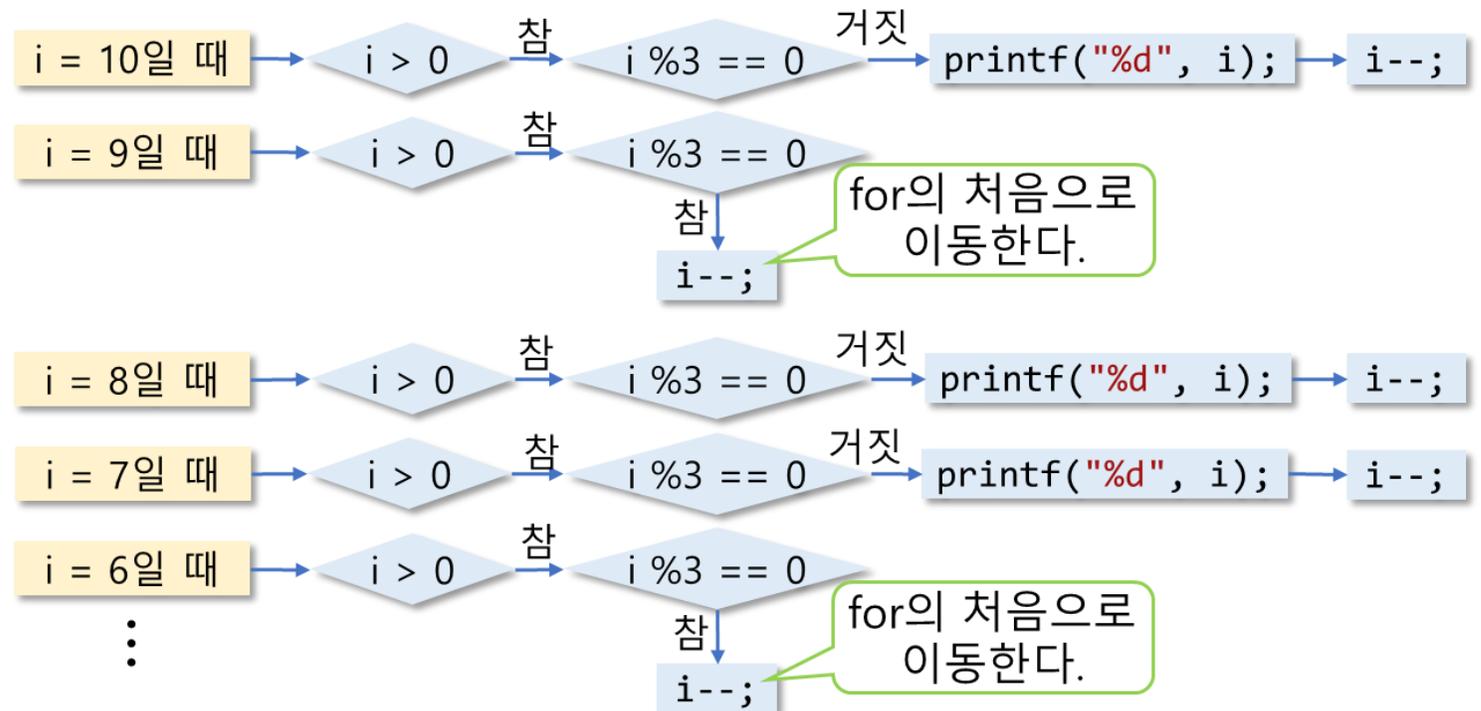
가 3의 배수일 때는 출력하지 않는다.

continue의 수행 순서

- for문 안에서 continue를 for의 시작 부분으로 이동해서 루프를 반복한다

```
for (i = 10; i > 0; i--)  
{  
    if (i % 3 == 0)  
        continue;  
    printf("%d ", i);  
}  
printf("\n");
```

10 8 7 5 4 2 1



goto문

- ❖ 제어의 흐름을 프로그램의 특정 위치로 이동
- ❖ 레이블 : 이동할 문장을 가리킨다
 - 레이블 이름과 콜론 사용
- ❖ 꼭 필요한 경우가 아니면 goto문을 사용하지 않는 것이 좋다
- ❖ 한꺼번에 여러 개의 루프를 탈출해야 할 때 유용

```
while (1)
{
    i = 10;
    while (i > 0) {
        if (i % 3 == 0)
            goto quit;
        printf("%d ", i--);
    }
}
quit:
printf("\n");
```

예제 : goto의 사용 예

```
03 int main(void)
04 {
05     int i;
06
07     for (i = 10; i > 0; i--)
08     {
09         if (i % 3 == 0)
10             goto quit; // quit 레이블이 지정하는 문장으로 이동한다.
11         printf("%d ", i);
12     }
13 quit:
14     printf("\n");
15
16     return 0;
17 }
```

실행결과

10

return문

- 프로그램 수행 중에 return문을 만나면 함수를 호출한 곳으로 되돌아간다
- 리턴 값이 있는 함수에서는 return 다음에 값을 써주고, 리턴 값이 없는 함수에서는 return만 써준다

```
void test(void)
{
    int i;
    for (i = 0; i < 10; i++)
    {
        if (i % 3 == 0)
            return; // 리턴 값이 없는 함수에서는 return만 써준다.
        printf("%d ", i);
    }
}
```

예제 : return의 사용 예

```
03 int main(void)
04 {
05     int i;
06
07     for (i = 10; i > 0; i--)
08     {
09         if (i % 3 == 0)
10             return 1;           // main 함수를 리턴시킨다. (프로그램 종료)
11         printf("%d ", i);
12     }
13     printf("\n");
14
15     return 0;
16 }
```

실행결과

10

추가자료

비만도 검사 프로그램

if~else 구문 사용

```
#include <stdio.h>
int main()
{
    double weight, height;
    double bmi;
    printf("몸무게를 입력하세요(kg) : ");
    scanf("%lf", &weight);
    printf("키를 입력하세요(cm) : ");
    scanf("%lf", &height);
    height = height / 100;
    bmi= weight / (height * height);
    printf("당신의BMI는: %.1lf입니다.\n", bmi);
    if(bmi>=20.0 && bmi<25.0)
        printf("표준체중입니다.\n");
    else
        printf("체중관리가필요합니다.\n");
    return 0;
}
```

/*BMI는 체중을 키의 제곱으로 나눈다 (체중은 kg, 키는 m)
BMI가 20미만: 저체중, 20~24.9 : 표준체중, 25이상: 과체중 */

학점계산 프로그램

if~else 구문 사용

```
#include <stdio.h>
int main()
{
    double avg;
    char grade;
    printf("평균점수를 입력하세요: ");
    scanf("%lf", &avg);

    if(avg >= 90.0) grade = 'A';
    else if(avg >= 80.0) grade = 'B';
    else if(avg >= 70.0) grade = 'C';
    else grade = 'F';

    printf("학점은 %c입니다.\n", grade);
    return 0;
}
```

```
/*평균점수는 실수 값으로 입력된다
평균이 90점 이상은 A, 90점 미만 80점 이상이면 B
80점 미만 70점 이상이면 C, 그 외는 F */
```

문자 종류별로 사용된 문자의 수를 세는 프로그램

```
#include <stdio.h>
int main(void)
{
    int blank_cnt = 0, c, digit_cnt = 0,
        letter_cnt = 0, nl_cnt = 0, other_cnt = 0;
    while ((c = getchar()) != EOF) /* braces not necessary */
        if (c == ' ') ++blank_cnt;
        else if (c >= '0' && c <= '9') ++digit_cnt;
        else if (c >= 'a' && c <= 'z' || c >= 'A' && c <= 'Z') ++letter_cnt;
        else if (c == '\n') ++nl_cnt;
        else ++other_cnt;
    printf("%10s%10s%10s%10s%10s%10s\n\n",
        "blanks", "digits", "letters", "lines", "others", "total");
    printf("%10d%10d%10d%10d%10d%10d\n\n",
        blank_cnt, digit_cnt, letter_cnt, nl_cnt, other_cnt,
        blank_cnt+ digit_cnt+ letter_cnt+ nl_cnt+ other_cnt);
    return 0;
}
```

구구단 프로그램

While 구문 사용

```
#include <stdio.h>
int main(void)
{
    int n;
    int i = 1;
    printf("출력하고 싶은 단: ");
    scanf("%d", &n);
    while(i <= 9)
    {
        printf("%d*%d = %d \n", n, i, n*i);
        i++;
    }
    return 0;
}
```

출력 결과

```
출력하고 싶은 단: 9
9*1=9
9*2=18
...
9*9=81
```

팩토리얼 프로그램

for 구문 사용

```
#include <stdio.h>
int main(void)
{
    long fact=1;
    int i, n;
    printf("정수를 입력하시오:");
    scanf("%d", &n);

    for(i = 1; i <= n; i++)
        fact = fact * i;

    printf("%d!은 %d입니다.\n", n, fact);
    return 0;
}
```

실행 결과

정수를 입력하시오: 10
10!은 3628800입니다.

추첨 프로그램

```
#include <stdio.h>
int main()
{
    int lot;
    printf("당첨 번호를 입력하세요: ");
    scanf("%d", &lot);
    switch(lot){
    case 1:
        printf("김서방"); break;
    case 2:
        printf("홍가네"); break;
    case 3:
        printf("안산택"); break;
    default:
        printf("당첨자가 없습니다!"); break;
    }
    return 0;
}
```

getchar()와 putchar()의 사용

❖stdio.h에 정의된 매크로

- getchar() : 키보드에서 문자를 읽음
- putchar() : 화면에 문자를 출력함

❖char 형은 1 바이트, int 형은 2 바이트나 4 바이트 에 저장됨

- int 형은 char 형이 저장할 수 있는 모든 값과 그 이상을 저장할 수 있음
- char 형을 작은 정수형으로 생각할 수 있음
- 반대로 int 형을 큰 문자형으로 볼 수 있음

예제-1

```
#include <stdio.h>
int main(void){
    int c;
    while ((c = getchar()) != EOF) {
        /* #define EOF (-1) */
        putchar(c);
    }
    return 0;
}
```

(주의) c가 **char**가 아니라 **int**로 선언되어 있음

예제-2

```
int main(){
    char str[80];
    int ch;
    int i;
    printf("문자열을 입력하세요 : ");
    for(i=0; i<4; i++){
        ch=getchar();
        str[i]=ch;
    }
    str[i]='\0';

    printf("입력된 문자열 : ");
    i=0;
    while(str[i]!='\0'){
        putchar(str[i]);
        i++;
    }
    return 0;
}
```

다

```
// 문자열을 저장할 배열
// getchar함수의 리턴값을 저장할 변수
// 반복 제어변수

// i는 0부터 3까지 변하면서 4번 반복
// 키보드로부터 문자를 입력 받아 리턴한다
// 리턴된 문자를 배열에 차례로 저장

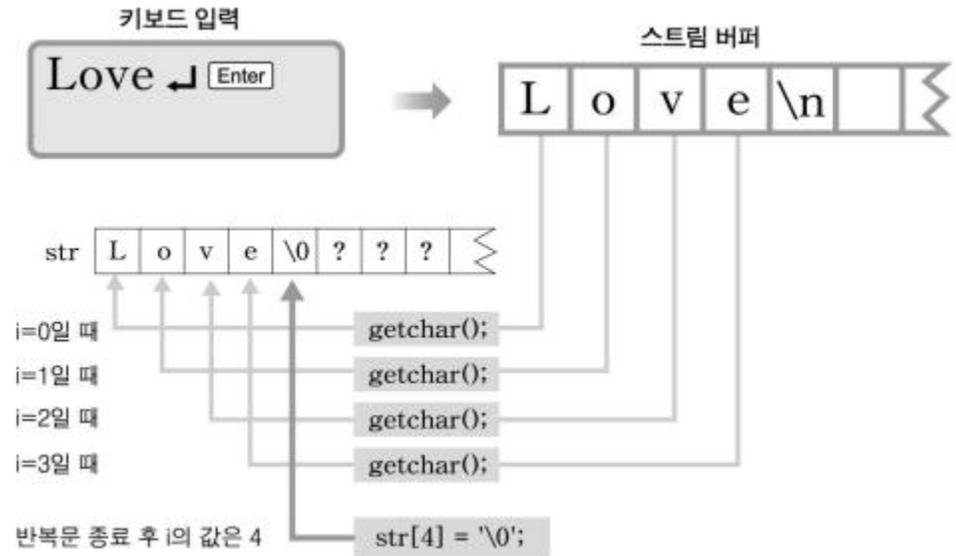
// 마지막에 널문자를 저장하여 문자열을 완성한

// 제어변수를 다시 0으로 초기화
// 배열요소가 널문자가 아닌 동안 반복
// 화면에 문자 출력
// 다음 문자로 이동
```

스트림 버퍼

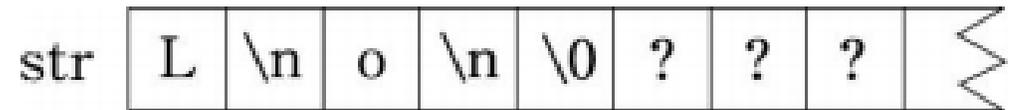
- 키보드에서 입력되는 데이터는 일단 버퍼에 저장되고 getchar함수는 버퍼로부터 데이터를 가져오므로 문자열은 한번에 입력한다

문자열을 입력하세요 : Love (엔터)
 입력된 문자열 : Love



- 문자를 하나씩 입력하면 새줄문자로 입력 되므로 문제가 발생한다

문자열을 입력하세요 : L (엔터)
 o (엔터) // 두 번째 문자를 입력할 때 입력이 끝난다.
 입력된 문자열 : L
 o



예제-3

- getchar함수가 새줄문자('\n')도 하나의 문자로 입력하므로 이 문자를 이용하여 한 줄을 입력 받을 수 있다

```
printf("문자열을 입력하세요 : ");
while(1){
    ch=getchar();
    if(ch=='\n') break;
        str[i]=ch;
        i++;
}
str[i]='\0';
```

// 무한 반복
// 퍼버로부터 문자 하나를 입력한다.
// 그 문자가 새줄문자이면 입력 종료
// 배열요소에 입력한 문자를 저장
// 다음 배열요소로 이동

// 마지막에 널문자를 넣어 문자열을 완성한다

키보드 입력

Love (엔터)



스트림 버퍼

L o v e \n

질의 및 응답

참고문헌

- 천정아, 『Core C Programming』, 연두에디션(2019)
- C가 보이는 그림책, ANK Co., Ltd. , 성안당 (2018)
- Greg Perry, Dean Miller 『어서와 C언어는 처음이지』, 천인국 옮김, 인피니티북스(2015)
- KELLEY (역 : 김명호 외), 『A Book on C』, 홍릉과학출판사 (2003)
- 윤성우, 『열혈 C 프로그래밍』, 오렌지미디어
- 천인국, 『쉽게 풀어쓴 C언어 Express』, 생능출판사
- 서현우, 『뇌를 자극하는 C 프로그래밍』, 한빛미디어
- 강성수, 『꽤도난마 C프로그래밍』, 북스홀릭
- 고응남, 『C프로그래밍 기초와 응용실습』, 정익사