
컴퓨터 프로그래밍 개론

조건문

목차

I. 순서도 개요

II. 조건문이란

III. if문

I. if

II. if - else

III. else if

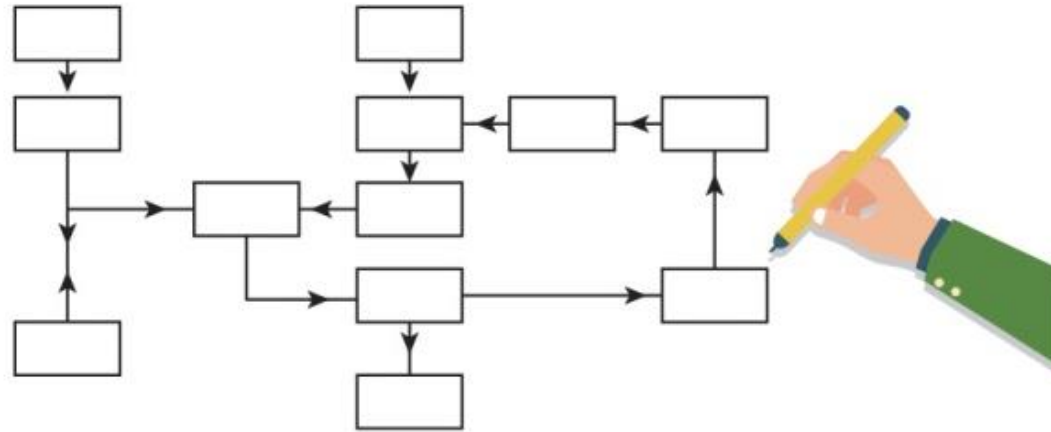
IV. 중첩 if문

IV. switch - case문

순서도

▶ 순서도(flow chart)

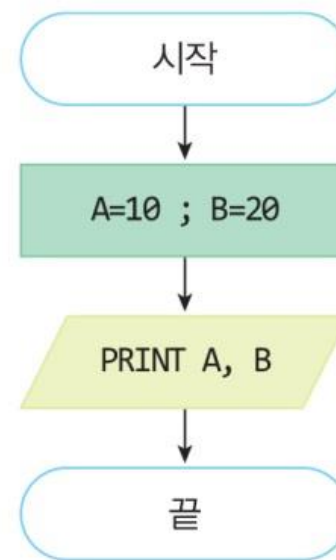
- ▶ 순서도는 일의 흐름을 나타내기 때문에 '흐름도'이라고 부르기도 하며 처리하려는 문제를 분석한 후 처리 순서를 단계적으로 표현
- ▶ 표준 기호와 연결선으로 작성하며 프로그래밍 과정에서는 프로그램의 설계도의 역할로 사용
- ▶ 순서도를 바탕으로 명확하고 논리적인 프로그램을 작성



순서도

▶ 순서도의 필요성

- ▶ 작업의 흐름을 도표로서 표시하므로 쉽게 이해
 - ▷ 간단한 도형 형태의 기호를 사용하여 간결한 도식화가 가능
- ▶ 특정 프로그래밍 언어에 관계없이 알고리즘의 표현이 가능
 - ▷ 프로그램 코딩의 기본 자료로서 활용이 가능
- ▶ 알고리즘의 이해와 추적이 쉬움
 - ▷ 문제 처리의 과정의 논리적 파악으로 잘못된 동작이나 에러의 검색이 쉬움



순서도

▶ 순서도의 기호와 의미

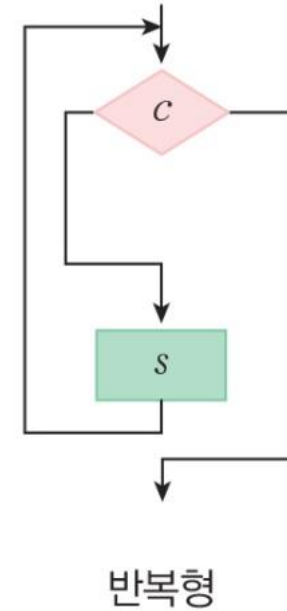
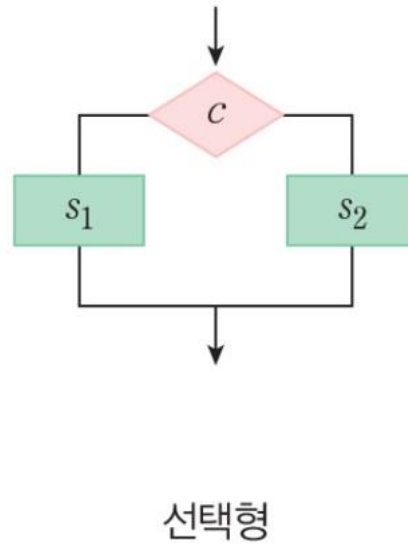
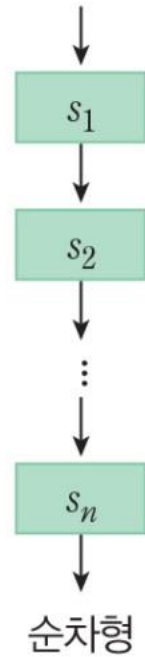
▶ 심벌의 형태, 이름, 의미를 이용하여 표현, 국제 표준

심벌 형태	이름	의미
	시작/끝 (start/stop)	순서도의 시작과 끝을 나타낸다.
	처리 (process)	지정된 동작, 연산, 값이나 데이터의 이동 등 모든 처리과정을 나타낸다.
	판단 (decision)	주어진 조건을 비교한 후 해당되는 조건에 따라 왼쪽 또는 오른쪽으로 흐름이 나누어지게 된다.
	입력/출력 (input/output)	일반적인 입력과 출력을 나타낸다.
	연결자 (connector)	흐름이 다른 곳으로 연결되거나 다른 곳으로부터 연결되는 입구를 나타낸다.
	프린트 (printing)	프린터를 통하여 출력한다.
	흐름선 (flow line)	명령어의 흐름을 나타내며, 화살표 방향의 순서대로 진행됨을 나타낸다,

순서도

▶ 순서도의 3가지 표현 방법

- ▶ 순차형(sequence) : 동작을 순서에 따라 차례로 실행
- ▶ 선택형(selection) : 조건에 맞는 동작 하나를 선택하여 실행
- ▶ 반복형(iteration) : 정해진 조건에 도달할 때까지 반복적인 동작을 수행, 일명 루프(loop)

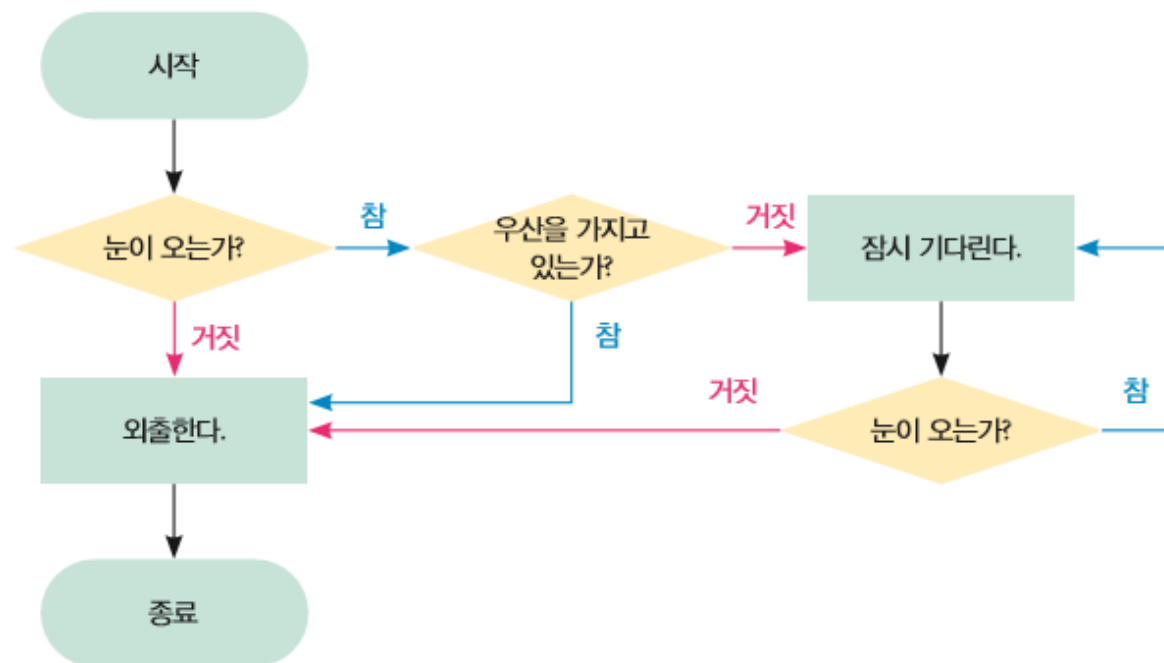


순서도

▶ 순서도 예제

▶ 눈이 올 때 어떻게 대처할 것인지에 대한 순서도

- ▷ 눈이 오지 않으면 외출
- ▷ 눈이 오면 우산을 가지고 있는지 검사
- ▷ 우산을 가지고 있다면 외출
- ▷ 우산을 가지고 있지 않다면 무한정 비가 그칠 때까지 기다림



조건문 개요

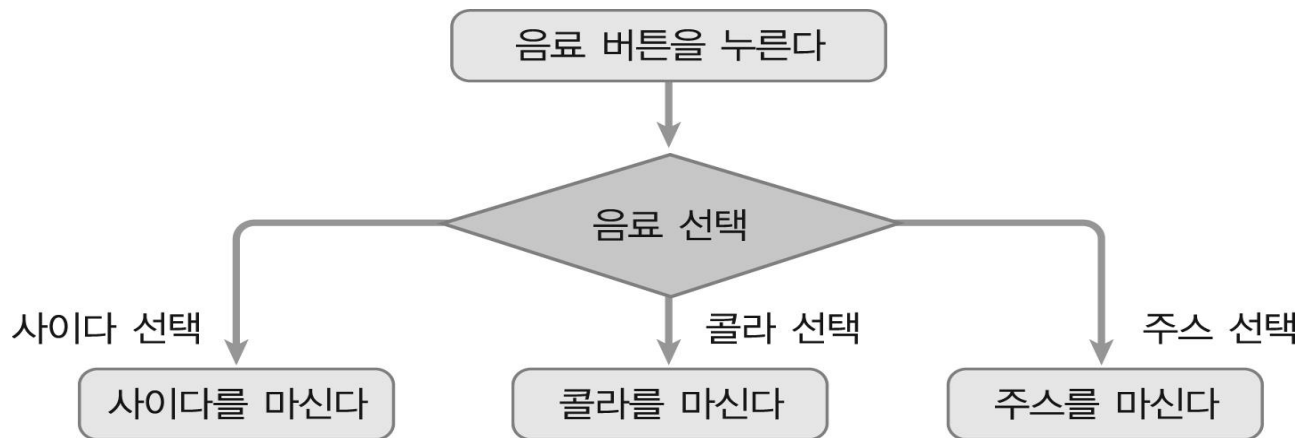
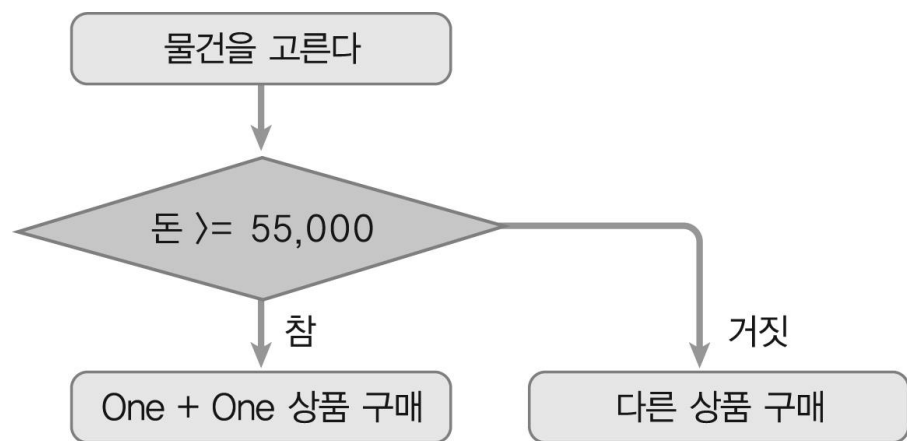
▶ 조건문이란

▶ 상황에 따라서 동작을 다르게 하는 프로그래밍 기법

▷ 비교 연산자($>=$, $<=$, \dots)나 논리 연산자($\&\&$, $\|\$, \dots)를 사용하여 논리값에 따라 다르게 실행

▶ 조건문은 프로그램의 흐름을 바꿀 수 있음

▶ 조건문은 선택의 기회를 제공할 수 있음



조건문 개요

▶ 조건문을 만드는 방법1 - if문

- ▶ if문
- ▶ if ~ else문
- ▶ 중첩 if ~ else문과 else if문

▶ 조건문을 만드는 방법2 - switch~case문

- ▶ 'else if문'을 간결하게 표현하였으며 가독성이 높음
- ▶ 그러나 'else if문'에 비하여 범위 지정 등의 제약이 있음

if문

▶ if 문 개요

▶ 조건의 참 거짓 여부에 따라 실행하거나 아무 동작도 하지 않음

▷ 즉, 조건식에 따라 실행문의 실행 여부를 결정

▶ 사용 방법

```
if (조건식){  
    실행문;  
}
```

▷ 조건식이 참이면 블록 내부의 실행문을 실행하고 그렇지 않으면 if문을 건너뛰고 다음 코드를 실행

▷ 실행문이 한 줄일 경우 블록을 생략할 수 있음

```
#include<stdio.h>  
int main(void){  
    int a = 20;  
    int b = 0;  
    if(a>10) b=a;  
    printf("a: %d, b: %d\n", a, b);  
    return 0;  
}
```

if문

▶ if 문의 형식

▶ if문은 중괄호{}를 사용하고 들여쓰기

▷ 실행문을 명확히 구분

▷ 중괄호는 실행문이 한 문장일 때는 생략 가능

▷ 아래의 예제는 모두 동일한 실행을 하는 정상적인 if 문

```
if(a > b)
{
    b = a;
}
```

```
if(a > b) {
    b = a;
}
```

```
if(a > b)
    b = a;
```

```
if(a > b) b = a;
```

실습문제 1

▶ if문을 이용하여 시험 점수가 80점이 이상이면 합격 판별을 하는 프로그램을 작성하시오.

▶ 출력결과

점수를 입력하시오 : 95
축하합니다. 합격입니다.
이 창을 닫으려면 아무 키나 누르세요.

점수를 입력하시오 : 79
이 창을 닫으려면 아무 키나 누르세요.

if - else문

▶ if ~ else문

- ▶ 서로 반대되는 조건을 한 번에 표현하거나 주어진 조건 이외의 상황에서 실행될 코드를 작성
 - ▷ 두 가지의 경우 중에 에 하나를 반드시 선택하는 경우 사용
- ▶ if문의 조건식이 참일 경우에는 else문 내부의 실행문은 동작하지 않음
 - ▷ 반대로 if문의 조건식이 거짓일 경우에는 반드시 else문의 내부에 있는 실행문이 처리됨
- ▶ else문은 홀로 쓰여질 수 없으며 반드시 if문과 함께 존재
- ▶ 사용 예제

```
#include<stdio.h>
int main(void)
{
    int a = -20;

    if(a>=0) a = 1;
    else a = -1;

    printf("a: %d\n",a);
    return 0;
}
```

if - else문

▶ 조건 연산자와 if - else문

- ▶ 조건 연산자 ? :는 if-else문으로 작성할 수 있음

```
i = a > b ? a - b : b - a;
```



```
if (a > b)
    i = a - b;
else
    i = b - a;
```

▶ if문과 if-else문

- ▶ 조건이 겹치지 않는다면 if문을 두 번 사용하는 것이 가능
- ▶ 두 번 검사하므로 속도 느림
- ▶ if문 두 번 사용 보다 if ~ else문을 사용하는 것이 효율적

기본 if문 두 문장 사용

```
if(a >= 0)
{
    a = 1;
}
if(a < 0)
{
    a = -1;
}
```



if ~ else문 한 문장 사용

```
if(a >= 0)
{
    a = 1;
}
else
{
    a = -1;
}
```

실습문제 2

▶ 키보드로 입력 받은 수가 3의 배수인지 판별하는 프로그램을 작성하시오.

▶ 출력 결과

양수를 입력하시오 : 129
3의 배수입니다.

양수를 입력하시오 : 128
3의 배수가 아닙니다.

if - else문

▶ if ~ else문의 한계

- ▶ 다양한 선택을 할 수 없기 때문에 여러 경우의 수를 요구하는 동작에는 사용할 수 없음
 - ▷ 참이나 거짓만 판단하기 때문
- ▶ 아래 코드는 사용자로부터 입력 받은 정수가 음수인지 양수인지 판별하는 코드
 - ▷ 0을 입력한다면?

```
#include <stdio.h>
int main(void)
{
    int num = 0;
    printf("숫자 입력 : ");
    scanf("%d", &num);

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

    else
        printf("음수입니다.\n");
    return 0;
}
```


중첩 if - else문

▶ 중첩 if - else문

- ▶ 조건문 내부에 조건문을 사용
- ▶ 앞서 배운 if문도 중첩하여 사용 가능

```
int main(void)
{
    int num = 0;
    printf("숫자 입력 : ");
    scanf("%d", &num);

    if(num>0)
        printf("양수입니다.\n");
    else {
        if(num==0)
            printf("0입니다.\n");

        else
            printf("음수입니다.\n");
    }
    return 0;
}
```

실습문제 3

▶ 중첩 if - else문을 사용하여 입력한 점수에 따라 성적을 출력하는 프로그램을 작성하시오.

▶ 정수로 입력 받으시오.

▶ A : 100 ~ 90

▶ B : 89 ~ 80

▶ ...

▶ ...

▶ F : 59 ~ 0

▶ 실행화면

점수를 입력하세요 : 89

A

점수를 입력하세요 : 66

D

else if문

▶ else if문

- ▶ 여러 개의 조건식을 사용할 수 있음
- ▶ 중첩 if - else문의 다른 표현

```
int main(void){
    int num = 0;
    printf("숫자 입력 : ");
    scanf("%d", &num);

    if(num>0)
        printf("양수입니다.\n");
    else {
        if(num==0)
            printf("0입니다.\n");

        else
            printf("음수입니다.\n");
    }

    return 0;
}
```

```
int main(void){
    int num = 0;
    printf("숫자 입력 : ");
    scanf("%d", &num);

    if(num>0)
        printf("양수입니다.\n");
    else if(num==0)
        printf("0입니다.\n");
    else
        printf("음수입니다.\n");

    return 0;
}
```

else if문

▶ else if문

▶ 중첩 if ~ else문과 else if문의 가독성 비교

```
if(num>=90)
    printf("A입니다.\n");
else
{
    if(num>=80)
        printf("B입니다.\n");
    else
    {
        if(num>=70)
            printf("C입니다.\n");
        else
        {
            if (num>=60)
                printf("D입니다. \n");
            else
                printf("F입니다. \n");
        }
    }
}
```

```
if(num>=90)
    printf("A입니다.\n");

else if(num>=80)
    printf("B입니다. \n");

else if(num>=70)
    printf("C입니다. \n");

else if (num>=60)
    printf("D입니다. \n");

else
    printf("F입니다. \n");
```

switch - case문

▶ switch - case 문 개요

▶ switch문은 조건식과 case 문의 상수식을 비교

▶ case의 비교 값과 일치하면 해당 case의 실행문장 수행

▶ 기본적으로 case는 break 포함하며 break를 만나면 switch문을 벗어남

▶ case의 비교 값과 일치하는 것이 없으면 default 문 실행

▶ 상수식에는 문자형 및 정수형 상수와 변수를 사용

▶ 문자열과 실수형 상수 및 실수형 변수를 사용할 수 없음

▶ break문

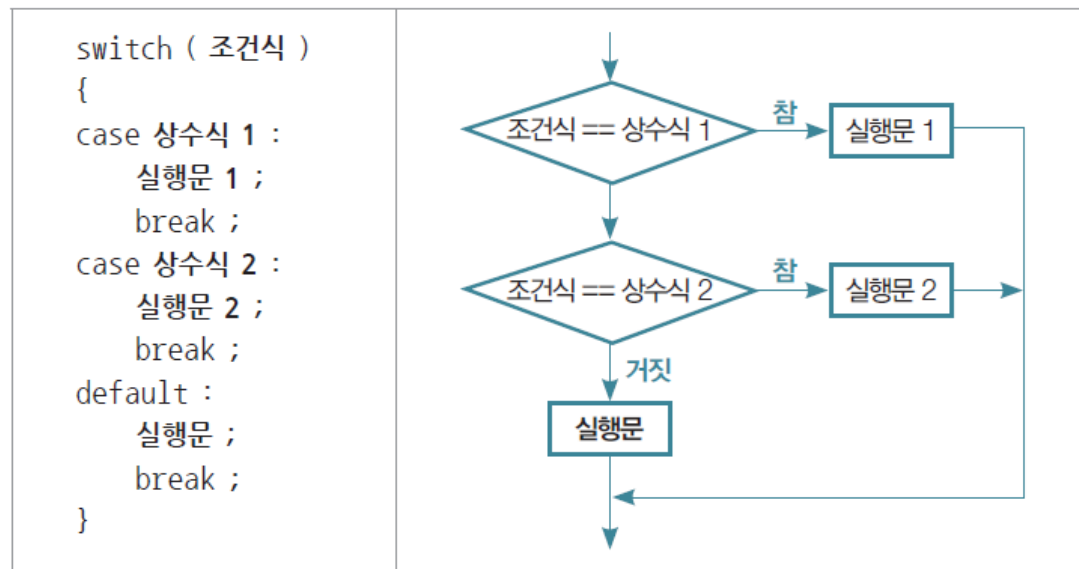
▶ switch문을 종료하는 역할

▶ break를 만나면 그 이후 내용은 무시되고 switch문 종료

▶ default문

▶ switch 문에서 정의한 case에 해당되는 조건이 없는 경우 수행

▶ default문은 생략 가능



switch - case문

▶ 입력한 문자에 따라서 요일을 출력하는 예제

```
#include <stdio.h>
int main(void)
{
    char ch;

    printf("(T)hursday, (F)riday, (S)aturday\n");
    printf("문자 입력(T, F, S) : ");
    scanf("%c", &ch);
```

```
(T)hursday, (F)riday, (S)aturday
문자 입력(T, F, S) : t
Thursday
-----
```

```
switch(ch)
{
    case 'T':
    case 't':
        printf("Thursday\n");
        break;
    case 'F':
    case 'f':
        printf("Friday\n");
        break;
    case 'S':
    case 's':
        printf("Saturday\n");
        break;
    default:
        printf("잘못 입력되었습니다.\n");
}
return 0;
}
```

switch - case문

▶ switch~case문과 if문

▶ 순위에 따라 상금을 결정하는 예제

```
switch (rank)
{
    case 1:
        m = 300;
        break;
    case 2:
        m = 200;
        break;
    case 3:
        m = 100;
        break;
    default:
        m = 10;
        break;
}
```



if문으로
바꿈

```
if (rank == 1)
{
    m = 300;
}
else if (rank == 2)
{
    m = 200;
}
else if (rank == 3)
{
    m = 100;
}
else
{
    m = 10;
}
```

실습문제 4

▶ 아래의 else if문을 switch - case문으로 변경하여 작성하시오.

▶ Hint : / 연산자의 특징

```
int score = 0;
printf("점수를 입력하세요 : ");
scanf("%d", &score);

if(num>=90)
    printf("A입니다.\n");
else if(num>=80)
    printf("B입니다. \n");
else if(num>=70)
    printf("C입니다. \n");
else if (num>=60)
    printf("D입니다. \n");
else
    printf("F입니다. \n");
```

Q & A