C언어 - HW2

임베디드스쿨1기
Lv1과정
2020. 08.01
손표훈

1. 연산자(Operator)

(1) 연산자란? 수학 또는 논리 기능을 수행하도록 하는 지시 기호
(2) 연사자의 종류는 아래 표와 같다

Operator	Description
+	덧셈
-	뺄셈
*	곱셈
/	나눗셈
%	나머지
++	1증가
--	1감소

표1. 산술연산자

Operator	Description
==	일치
!=	불일치
>	좌항이 크다
<	좌항이 작다
>=	좌항이 크거나 같다
<=	좌항이 작거나 같다

표2. 관계연산자

-> ++, -- 같은 증감 연산자는 변수 앞/뒤 위치에 따라 결과 값이 달라진다.
num = 2; 일 때,
printf("num = %d\n", num++); or printf("num = %d\n", num++);
num = 2 num = 3

1. 연산자(Operator)

Operator	Description
&&	AND
||	OR
!	NOT

표3. 논리 연산자

Operator	Description
&	AND
|	OR
^	XOR
~	Inverter
<<	좌 시프트
>>	우 시프트

표4. 비트 연산자

Operator	Description
=	좌항에 대입
+=	우항+좌항을 좌항에 대입
-=	우항-좌항을 좌항에 대입
*=	우항x좌항을 좌항에 대입
/=	좌항/우항을 좌항에 대입
%=	좌항%우항을 좌항에 대입
<<=	우항 수만큼 좌시프트 후 좌항에 대입
>>=	우항 수만큼 우시프트 후 좌항에 대입
&=	좌 AND 우 후 좌항에 대입
^=	좌 XOR 우 후 좌항에 대입
|=	좌 OR 우 후 좌항에 대입

표5. 대입 연산자

1. 연산자(Operator)

(3) 산술연산자

```c
#include <stdio.h>

int main(void)
{
    int a = 21;
    int b = 10;
    int c;

    c = a+b;

    printf("Line 1 - Value of C is %d\n", c);

    c = a-b;

    printf("Line 2 - Value of C is %d\n", c);

    c = a*b;

    printf("Line 3 - Value of C is %d\n", c);

    c = a/b;

    printf("Line 4 - Value of C is %d\n", c);

    c = a%b;

    printf("Line 5 - Value of C is %d\n", c);

    c = a++;

    printf("Line 6 - Value of C is %d\n", c);

    c = a--;

    printf("Line 7 - Value of C is %d\n", c);

    return 0;
}
```

```
Line 1 - Value of C is 31
Line 2 - Value of C is 11
Line 3 - Value of C is 210
Line 4 - Value of C is 2
Line 5 - Value of C is 1
Line 6 - Value of C is 21
Line 7 - Value of C is 22
```

1. 연산자(Operator)

(4) 관계연산자

```c
#include <stdio.h>

int main(void)
{
    int a = 21;
    int b = 10;
    int c;

    if(a==b)
    {
        printf("Line 1 - a is equal to b\n");
    }
    else
    {
        printf("Line 1 - a is not equal to b\n");
    }
    if(a<b)
    {
        printf("Line 2 - a is less than b\n");
    }
    else
    {
        printf("Line 2 - a is not less than b\n");
    }
    if(a>b)
    {
        printf("Line 3 - a is greater than b\n");
    }
    else
    {
        printf("Line 3 - a is not greater than b\n");
    }

    a = 5;
    b = 20;

    if(a<=b)
    {
        printf("Line 4 - a is either less than or equal to b\n");
    }
    if(a>=b)
    {
        printf("Line 5 - b is either greater than or equal to a\n");
    }

    return 0;
}
```

```
Line 1 - a is not equal to b
Line 2 - a is not less than b
Line 3 - a is greater than b
Line 4 - a is either less than or equal to b
```

1. 연산자(Operator)

(5) 논리연산자

```c
#include <stdio.h>

int main(void)
{
    int a = 5;
    int b = 20;
    int c;

    if(a&&b)
    {
        printf("Line 1 - Condition is true\n");
    }

    if(a||b)
    {
        printf("Line 2 - Condition is true\n");
    }

    a = 0;
    b = 10;

    if(a&&b)
    {
        printf("Line 3 - Condition is true\n");
    }
    else
    {
        printf("Line 3 - Condition is not true\n");
    }

    if(!(a&&b))
    {
        printf("Line 4 - Condition is true\n");
    }

    return 0;
}
```

```
Line 1 - Condition is true
Line 2 - Condition is true
Line 3 - Condition is not true
Line 4 - Condition is true
```

1. 연산자(Operator)

(6) 비트연산자

```c
#include <stdio.h>

int main(void)
{
    unsigned int a = 60;
    unsigned int b = 13;
    int c = 0;

    c = a&b;
    printf("Line 1 - Value of C is %d\n", c);

    c = a|b;
    printf("Line 2 - Value of C is %d\n", c);

    c = a^b;
    printf("Line 3 - Value of C is %d\n", c);

    c = ~a;
    printf("Line 4 - Value of C is %d\n", c);

    c = a<<2;
    printf("Line 5 - Value of C is %d\n", c);

    c = a>>2;
    printf("Line 6 - Value of C is %d\n", c);

    return 0;
}
```

```
Line 1 - Value of C is 12
Line 2 - Value of C is 61
Line 3 - Value of C is 49
Line 4 - Value of C is -61
Line 5 - Value of C is 240
Line 6 - Value of C is 15
```

1. 연산자(Operator)

(7) 대입연산자

```c
#include <stdio.h>

int main(void)
{
    int a = 21;
    int c;

    c = a;
    printf("Line 1 - = Operator Example, Value of C = %d\n", c);
    c += a;
    printf("Line 2 - += Operator Example, Value of C = %d\n", c);
    c -= a;
    printf("Line 3 - -= Operator Example, Value of C = %d\n", c);
    c *= a;
    printf("Line 4 - *= Operator Example, Value of C = %d\n", c);
    c /= a;
    printf("Line 5 - /= Operator Example, Value of C = %d\n", c);

    c = 200;
    c %= a;
    printf("Line 6 - %%= Operator Example, Value of C = %d\n", c);
    c <<= 2;
    printf("Line 7 - <<= Operator Example, Value of C = %d\n", c);
    c >>= 2;
    printf("Line 8 - >>= Operator Example, Value of C = %d\n", c);
    c &= 2;
    printf("Line 9 - &= Operator Example, Value of C = %d\n", c);
    c ^= 2;
    printf("Line 10 - ^= Operator Example, Value of C = %d\n", c);
    c |= 2;
    printf("Line 11 - |= Operator Example, Value of C = %d\n", c);

    return 0;
}
```

```
Line 1 - = Operator Example, Value of C = 21
Line 2 - += Operator Example, Value of C = 42
Line 3 - -= Operator Example, Value of C = 21
Line 4 - *= Operator Example, Value of C = 441
Line 5 - /= Operator Example, Value of C = 21
Line 6 - %= Operator Example, Value of C = 11
Line 7 - <<= Operator Example, Value of C = 44
Line 8 - >>= Operator Example, Value of C = 11
Line 9 - &= Operator Example, Value of C = 2
Line 10 - ^= Operator Example, Value of C = 0
Line 11 - |= Operator Example, Value of C = 2
```

2. 의사결정(Decision Making)

(1) 의사결정은 특정 조건에 따라 문장의 실행 순서를 결정하거나 특정 조건이 충족 할 때 까지 문장 그룹을 반복(Branch명령어)

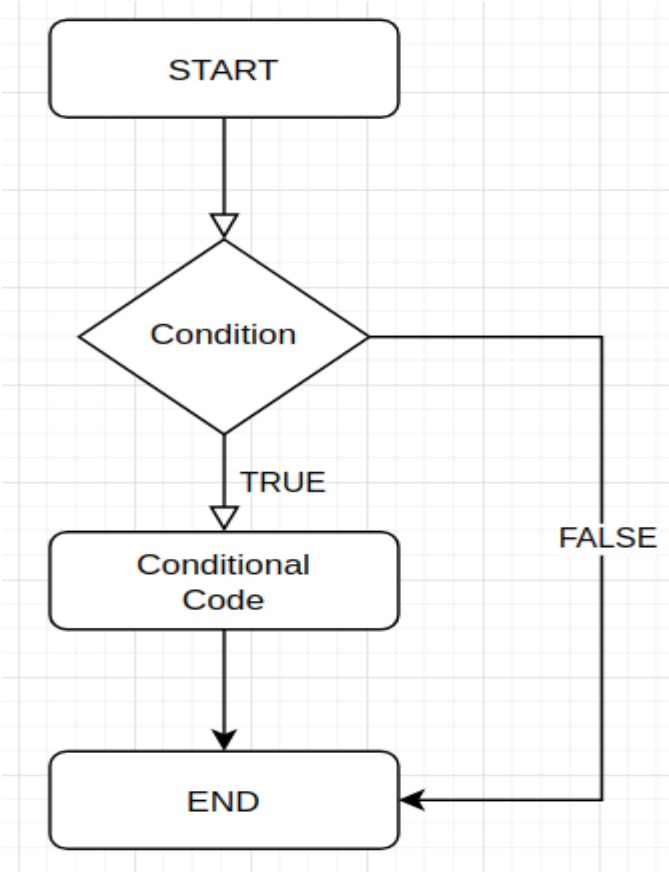


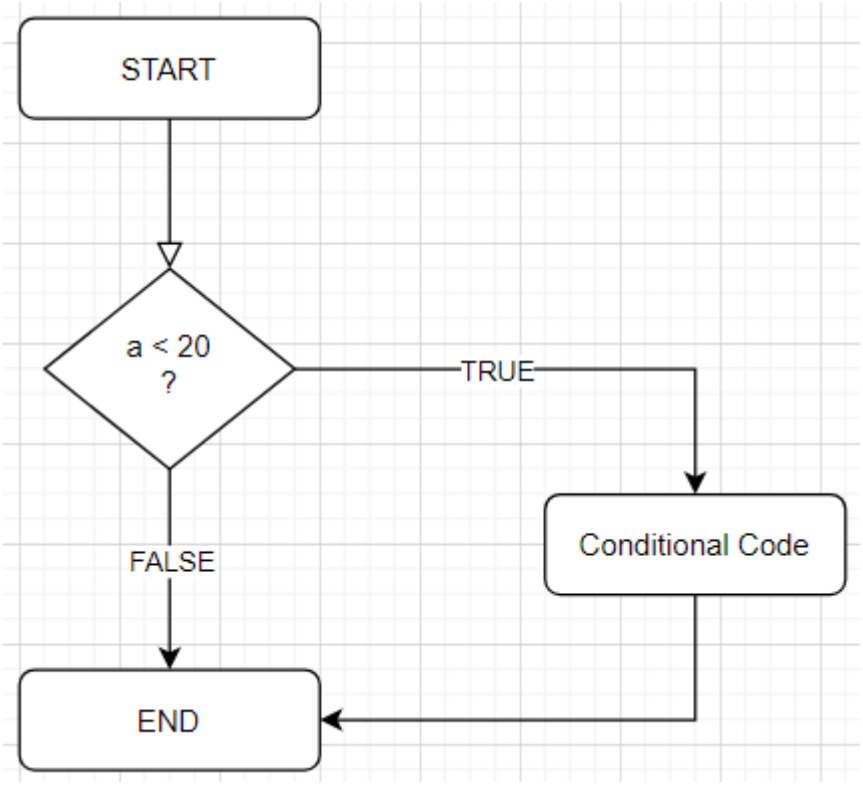
그림1. 조건문 실행 순서도

2. 의사결정(Decision Making)

(2) IF

```c
#include <stdio.h>

int main(void)
{
        int a = 10;

        if(a<20)
        {
                printf("a is less than 20\n");
        }

        printf("Value of a is : %d\n", a);

        return 0;
}
```

```
a is less than 20
Value of a is : 10
```

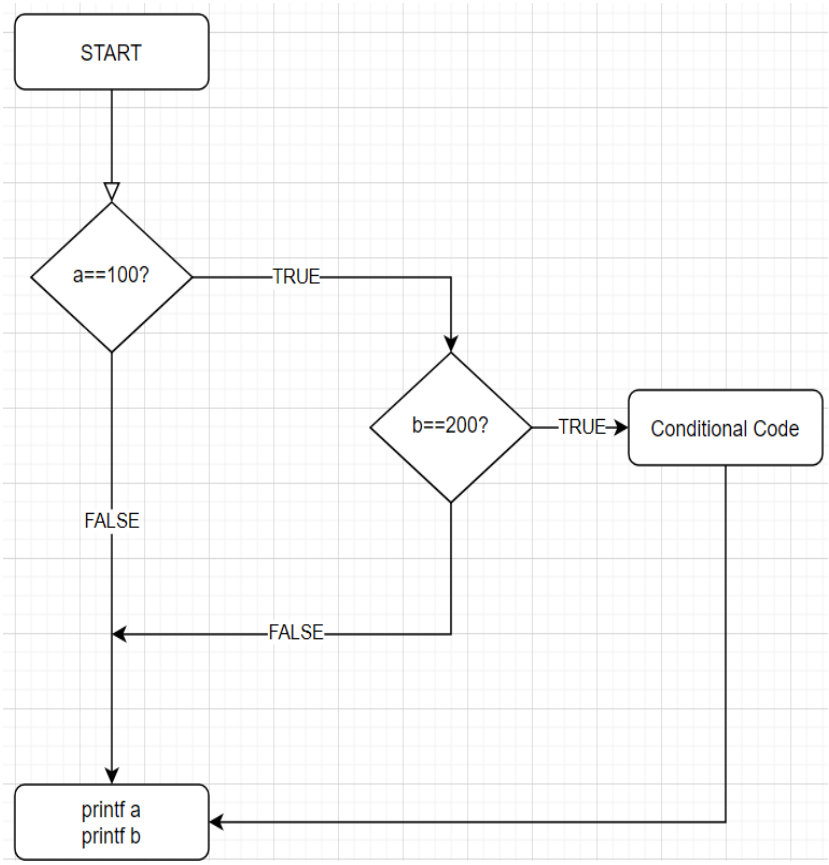


2. 의사결정(Decision Making)

(3) IF2

```c
#include <stdio.h>

int main(void)
{
        int a = 100;
        int b = 200;

        if(a == 100)
        {
                if(b == 200)
                {
                        printf("Value of a is 100 and b is 200\n");
                }
        }

        printf("Exact value of a is : %d\n", a);
        printf("Exact value of b is : %d\n", b);

        return 0;
}
```

```
Value of a is 100 and b is 200
Exact value of a is : 100
Exact value of b is : 200
```

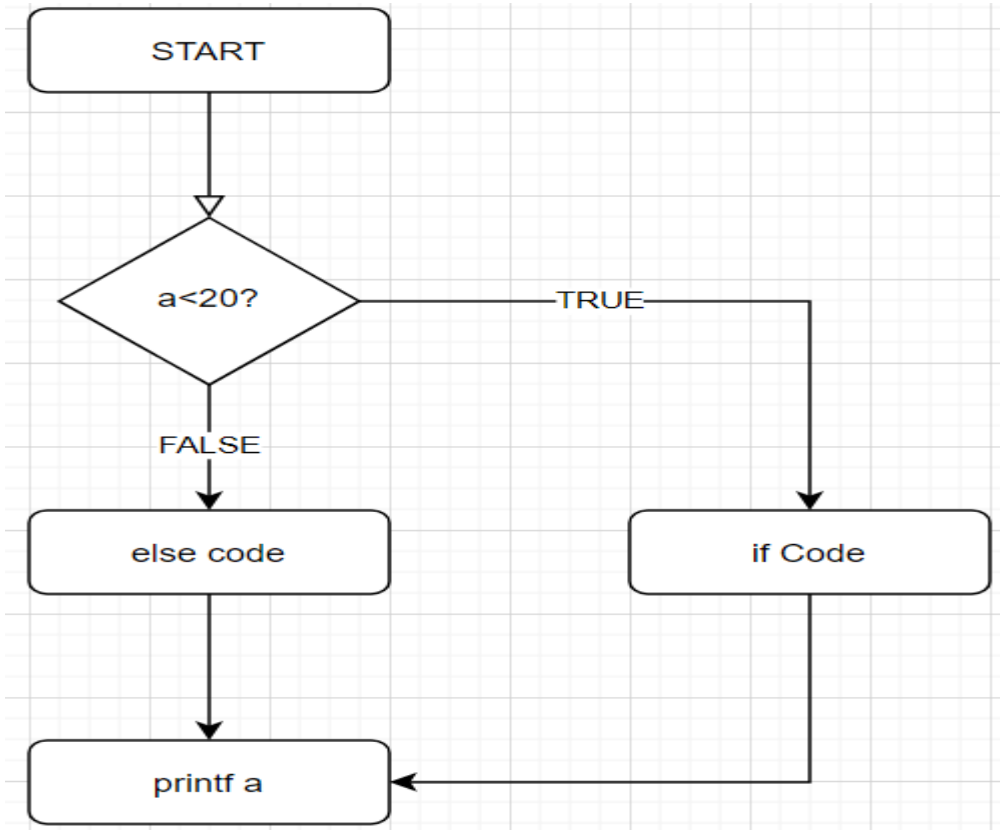


2. 의사결정(Decision Making)

(3) if-else

```c
#include <stdio.h>

int main(void)
{
        int a = 100;

        if(a < 20)
        {
                printf("a is less than 20\n");
        }
        else
        {
                printf("a is not less than 20\n");
        }

        printf("value of a is : %d\n", a);

        return 0;
}
```

```
a is not less than 20
value of a is : 100
```

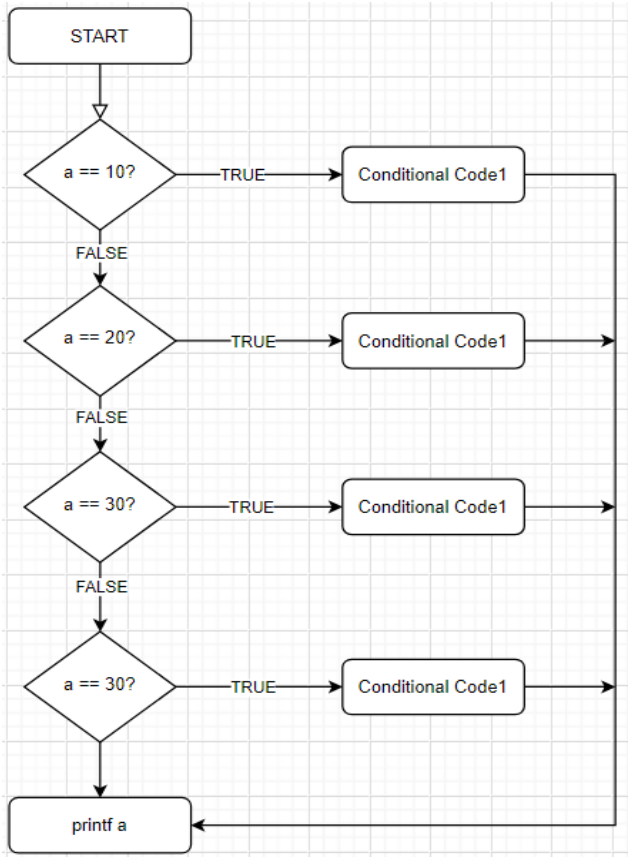


2. 의사결정(Decision Making)

(4) if-else if

```c
#include <stdio.h>

int main(void)
{
        int a = 100;

        if(a == 10)
        {
                printf("Value of a is 10\n");
        }
        else if(a == 20)
        {
                printf("Value of a is 20\n");
        }
        else if(a == 30)
        {
                printf("Value of a is 30\n");
        }
        else
        {
                printf("None of the value is matching\n");
        }

        printf("Exact value of a is : %d\n", a);

        return 0;
}
```

```
None of the value is matching
Exact value of a is : 100
```

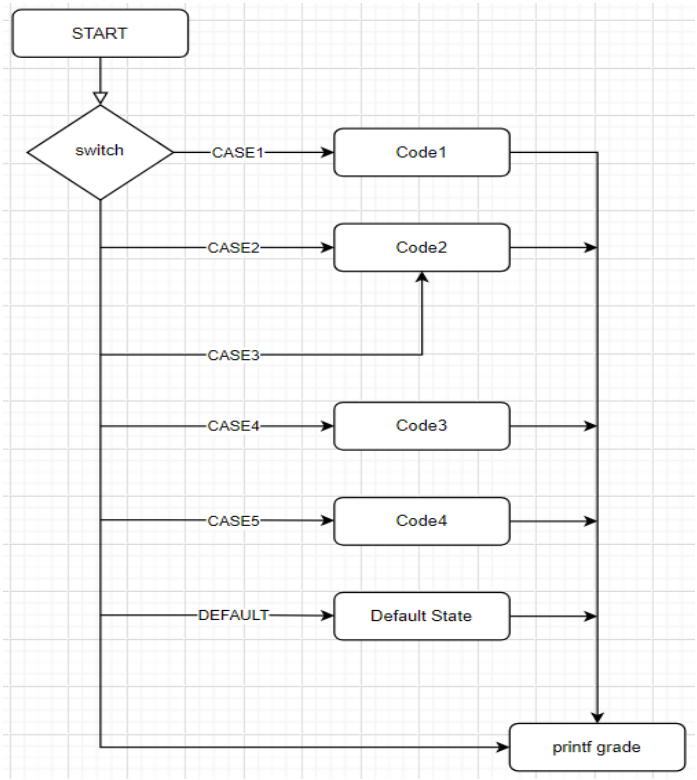


2. 의사결정(Decision Making)

(5) Switch

```c
#include <stdio.h>

int main(void)
{
        char grade = 'B';

        switch(grade)
        {
                case 'A':
                        printf("Excellent!\n");
                        break;
                case 'B':
                case 'C':
                        printf("Well done!\n");
                        break;
                case 'D':
                        printf("You passed!\n");
                        break;
                case 'F':
                        printf("Better try again!\n");
                        break;
                default:
                        printf("Invalid grade\n");
        }

        printf("Your grade is %c\n", grade);

        return 0;
}
```

```
Well done!
Your grade is B
```

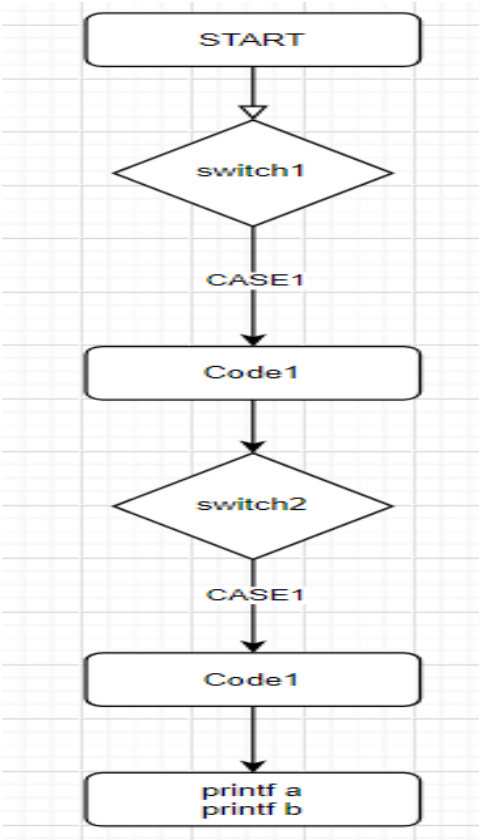


2. 의사결정(Decision Making)

(6) Switch2

```c
#include <stdio.h>

int main(void)
{
        int a = 100;
        int b = 200;

        switch(a)
        {
                case 100:
                        printf("This is part of outer switch\n");

                        switch(b)
                        {
                                case 200:
                                        printf("This is part of inner switch\n");
                        }
        }

        printf("Exact value of a is : %d\n", a);
        printf("Exact value of b is : %d\n", b);
        return 0;
}
```

```
This is part of outer switch
This is part of inner switch
Exact value of a is : 100
Exact value of b is : 200
```



3. 루프제어(Loop Control)

1) 반복문 : 특정 조건에 도달할 때 까지 반복되는 명령어

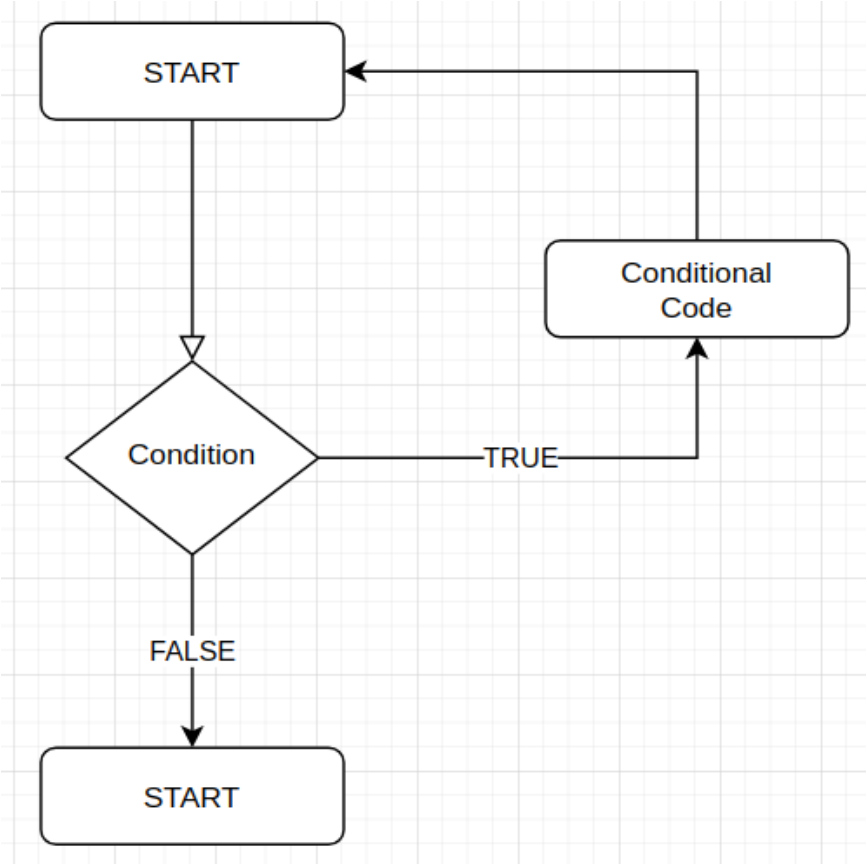
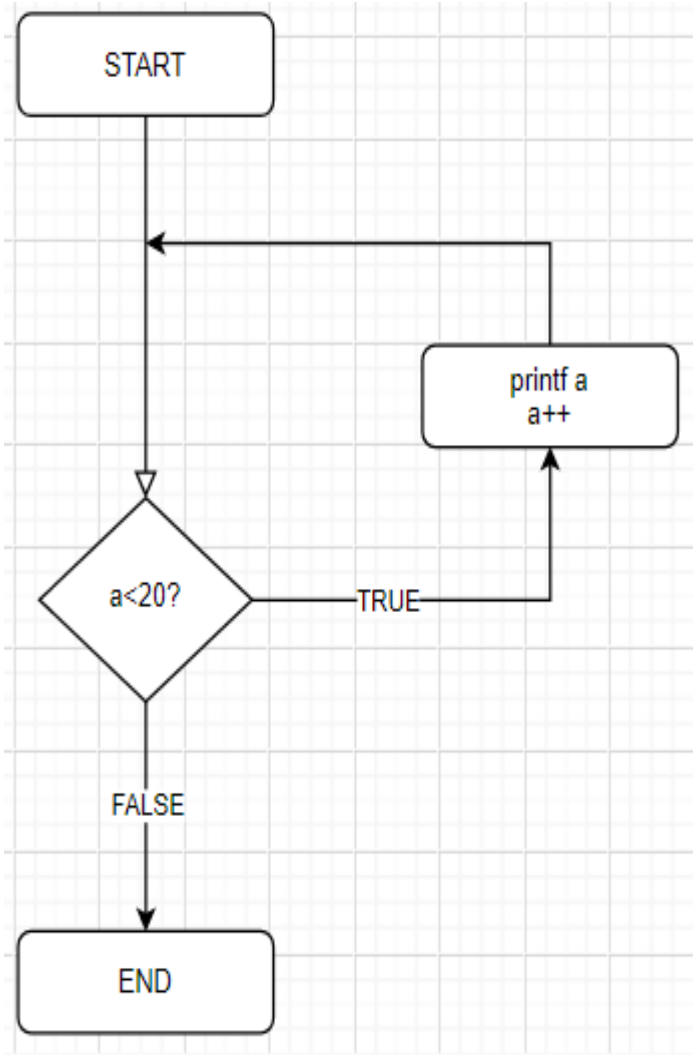


그림1. 반복문 실행 순서도

3. 루프제어(Loop Control)

(1) While

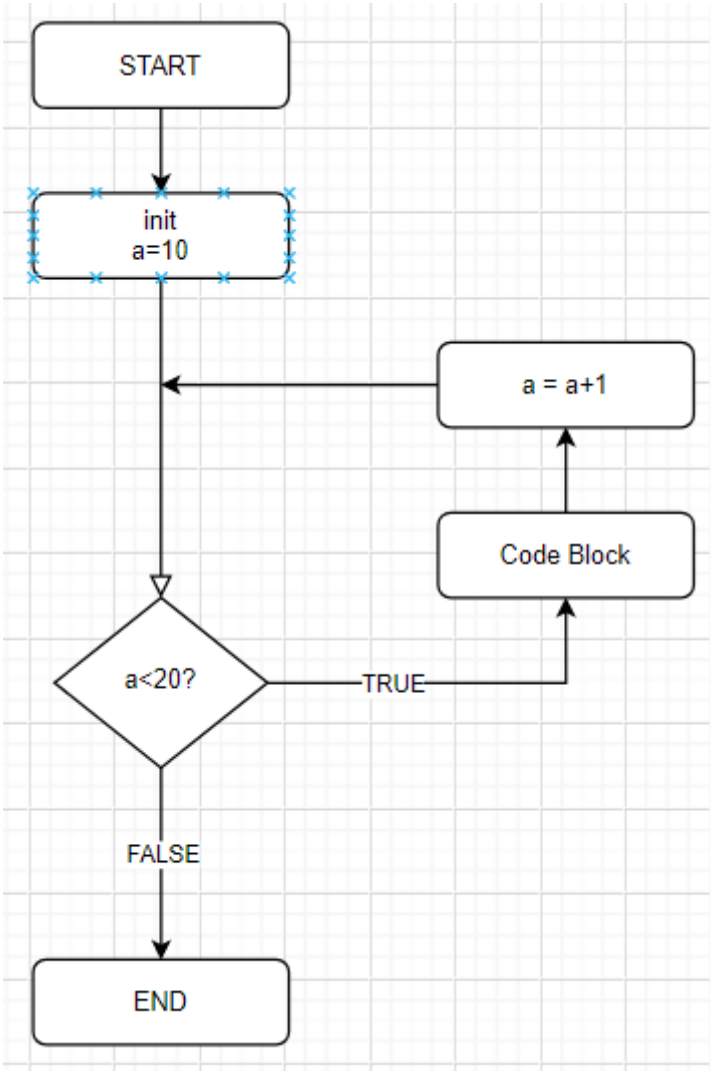



```c
#include <stdio.h>

int main(void)
{
        int a = 10;

        while(a < 20)
        {
                printf("value of a : %d\n", a);
                a++;
        }
        return 0;
}
```

```
value of a : 10
value of a : 11
value of a : 12
value of a : 13
value of a : 14
value of a : 15
value of a : 16
value of a : 17
value of a : 18
value of a : 19
```

3. 루프제어(Loop Control)

(2) for

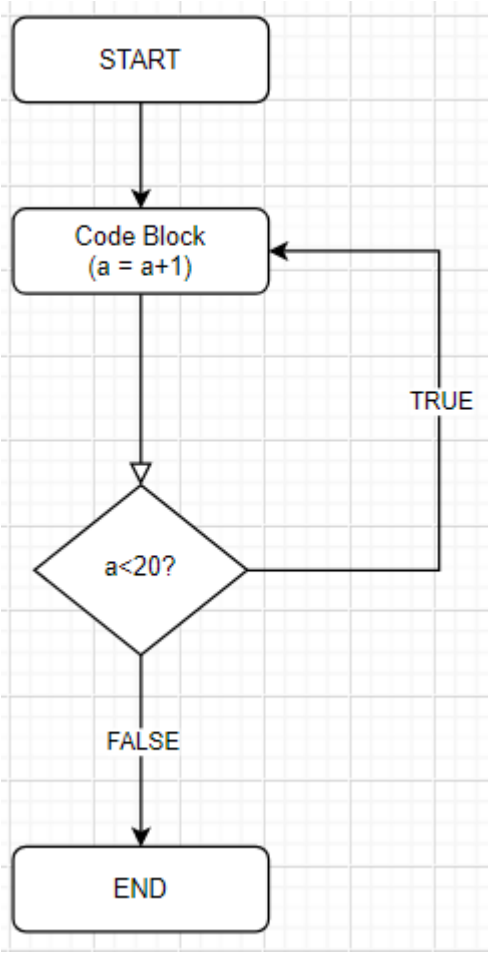



```c
#include <stdio.h>

int main(void)
{
        int a;

        for(a = 10; a < 20; a = a+1)
        {
                printf("value of a : %d\n", a);
        }
        return 0;
}
```

```
value of a : 10
value of a : 11
value of a : 12
value of a : 13
value of a : 14
value of a : 15
value of a : 16
value of a : 17
value of a : 18
value of a : 19
```

3. 루프제어(Loop Control)

(3) do-while

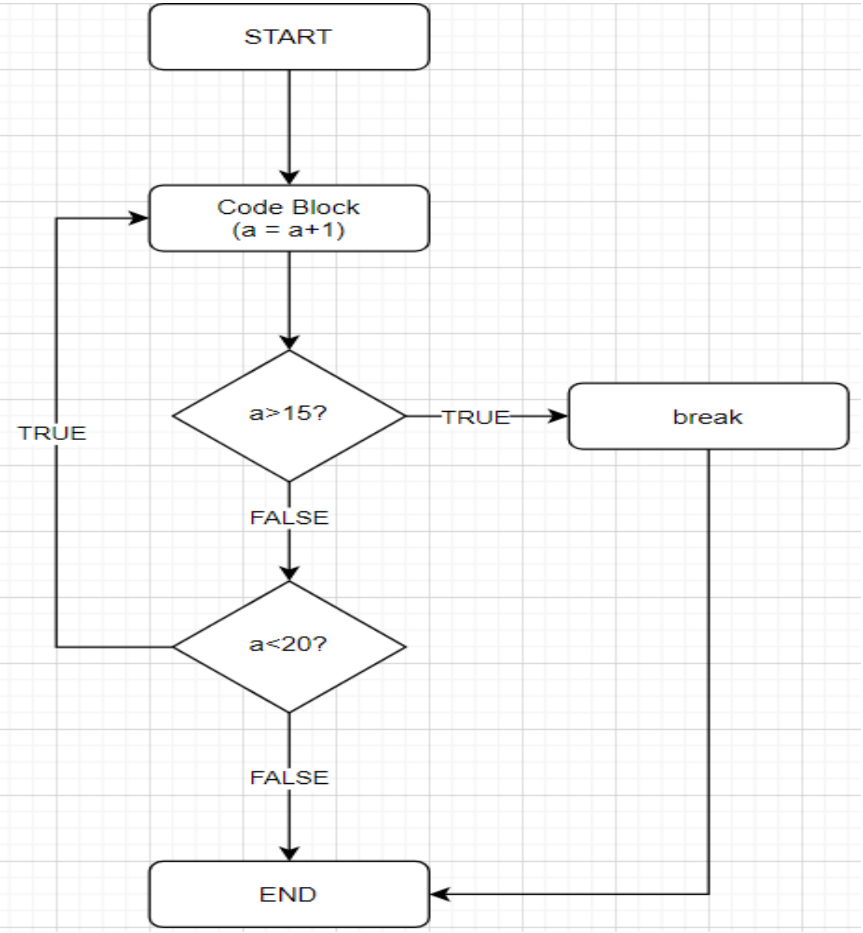



```c
#include <stdio.h>

int main(void)
{
        int a = 10;

        do
        {
                printf("value of a : %d\n", a);
                a = a+1;
        }while(a < 20);

        return 0;
}
```

```
value of a : 10
value of a : 11
value of a : 12
value of a : 13
value of a : 14
value of a : 15
value of a : 16
value of a : 17
value of a : 18
value of a : 19
```

3. 루프제어(Loop Control)

(4) Break

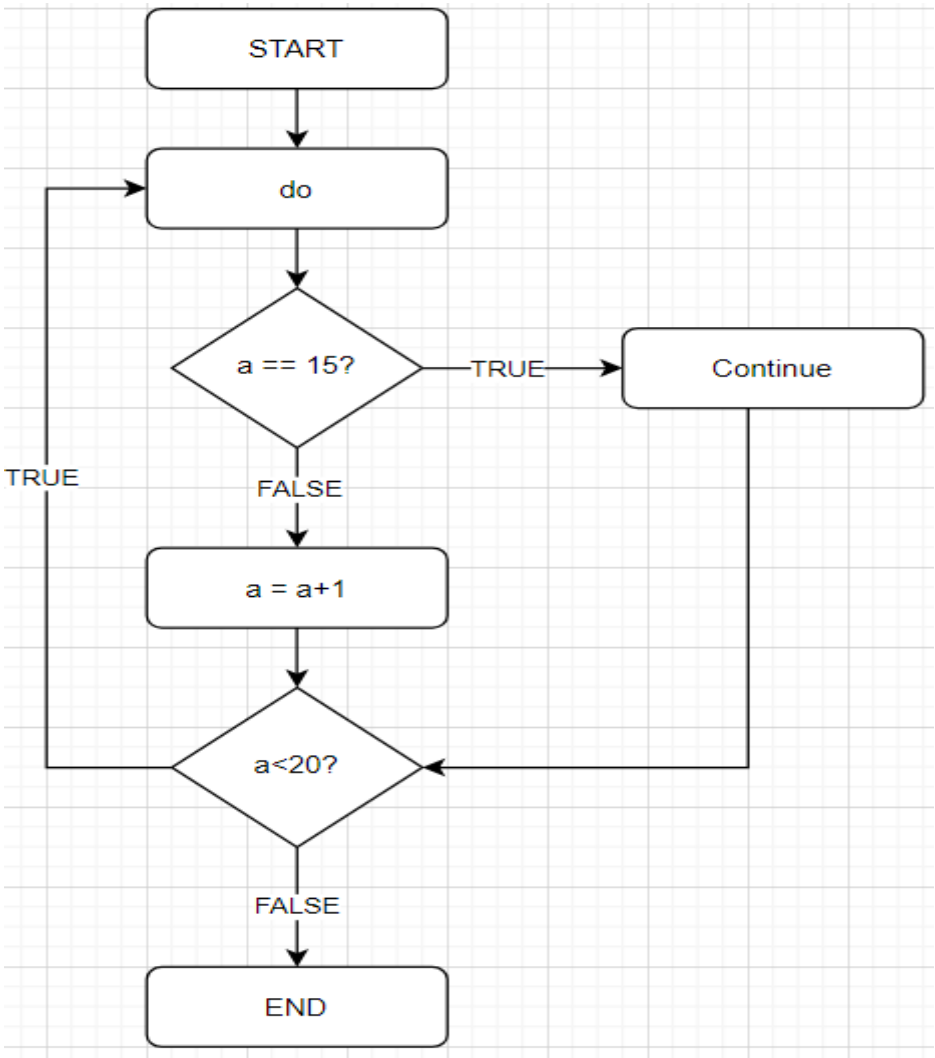



```c
#include <stdio.h>

int main(void)
{
        int a = 10;

        while(a < 20)
        {
                printf("value of a : %d\n", a);
                a++;

                if(a > 15)
                {
                        break;
                }
        }

        return 0;
}
```

```
value of a : 10
value of a : 11
value of a : 12
value of a : 13
value of a : 14
value of a : 15
```

3. 루프제어(Loop Control)

(5) Continue

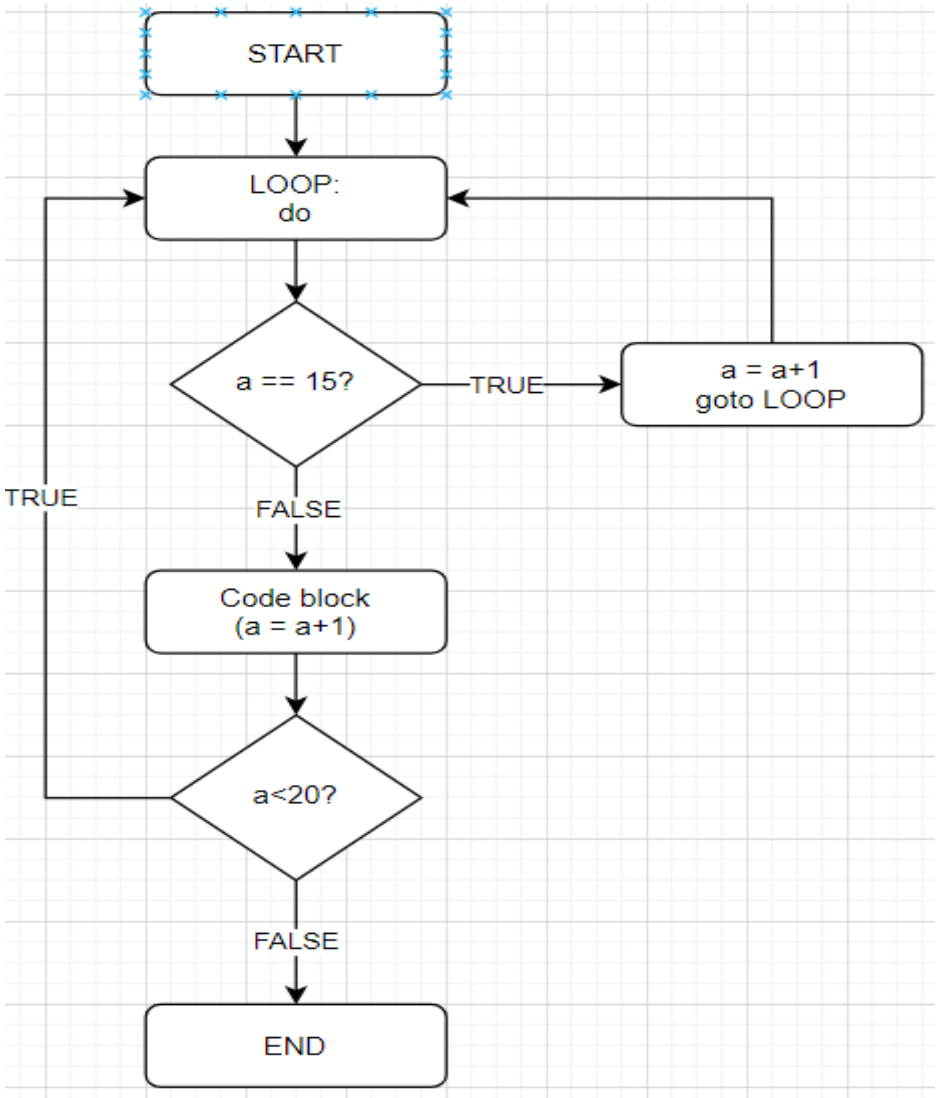



```c
#include <stdio.h>

int main(void)
{
        int a = 10;

        do
        {
                if(a == 15)
                {
                        a = a+1;
                        continue;
                }

                printf("value of a : %d\n", a);
                a++;

        }while(a < 20);

        return 0;
}
```

```
value of a : 10
value of a : 11
value of a : 12
value of a : 13
value of a : 14
value of a : 16
value of a : 17
value of a : 18
value of a : 19
```

3. 루프제어(Loop Control)

1) 반복문 : 특정 조건에 도달할 때 까지 반복되는 명령어




```c
#include <stdio.h>

int main(void)
{
        int a = 10;

LOOP:do
        {
                if(a == 15)
                {
                        a = a+1;
                        goto LOOP;
                }

                printf("value of a : %d\n", a);
                a++;
        }while(a < 20);

        return 0;
}
```

```
value of a : 10
value of a : 11
value of a : 12
value of a : 13
value of a : 14
value of a : 16
value of a : 17
value of a : 18
value of a : 19
```