生態模擬: 以C語言為例

Class 05 (2018/04/12)

Control Structure (控制結構)

[條件執行] Condition

- 5.1 Relational operators and conditions (關係運算子)
- 5.2 If statement
- 5.3 If … else statement
- 5.4 If … else if … else statement
- 5.5 switch statement
- 5.6 Logical operators (邏輯運算子)

[迴路] Loop

- 5.7 for loop
- 5.8 do…while loop
- 5.9 Nested structure of statements
- 5.10 break and continue statement

Takeshi Miki

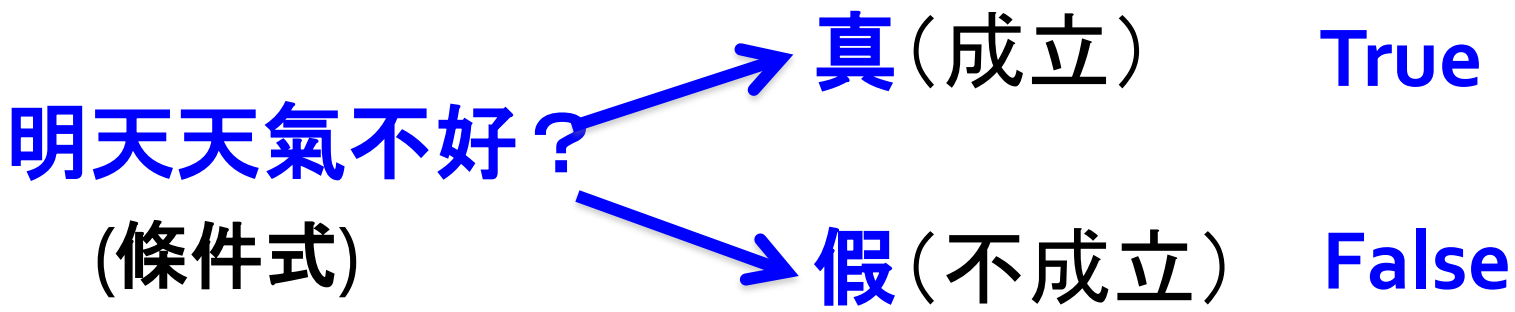
三木　健（海洋研究所）

5.1 Relational operators and conditions (關係運算子 &條件式)

What is a condition?

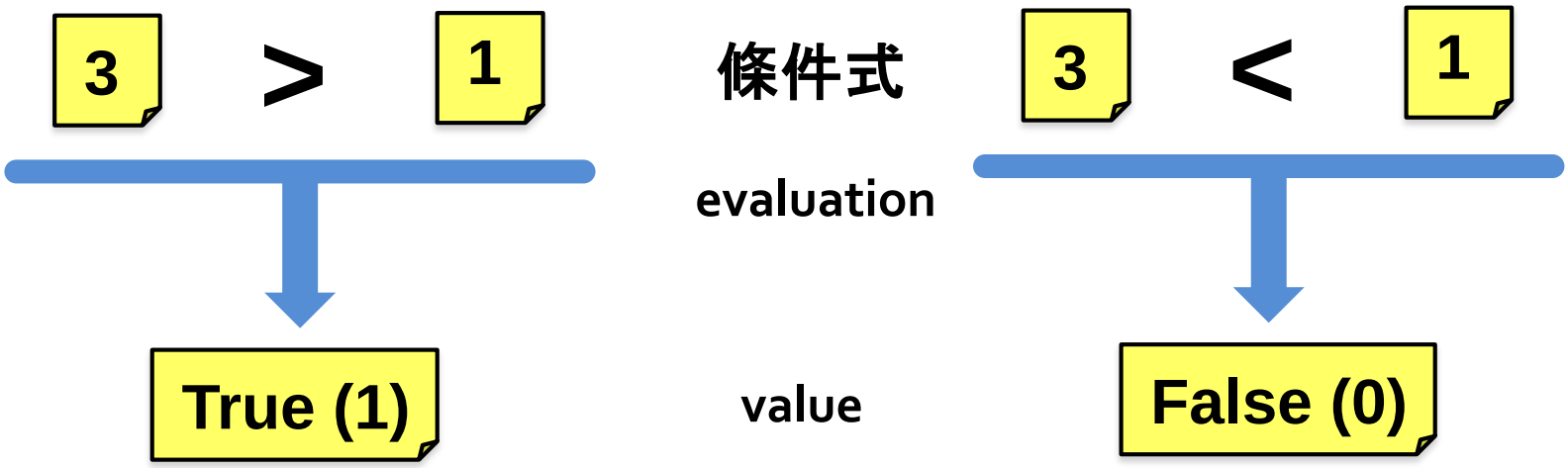
如果明天天氣不好，我不想去大學。

There are only two exclusive possibilities.



5.1 Relational operators and conditions (關係運算子 & 條件式)

How to write a condition?



5.1 Relational operators and conditions (關係運算子 & 條件式)

Relational operators (>, >=, <, <=, ==, !=)

Prepare a new sample file of C codes.

```
/**/
#include <stdio.h>

int main (void)
{
        printf("3 > 1\t%d\n", 3 > 1);
        printf("3 >= 1\t%d\n", 3 >= 1);
        printf("3 < 1\t%d\n", 3 < 1);
        printf("3 <= 1\t%d\n", 3 <= 1);
        printf("3 == 1\t%d\n", 3 == 1);
        printf("3 != 1\t%d\n", 3 != 1);

        return 0;
}
```

= (代入演算子)

== (關係運算子)

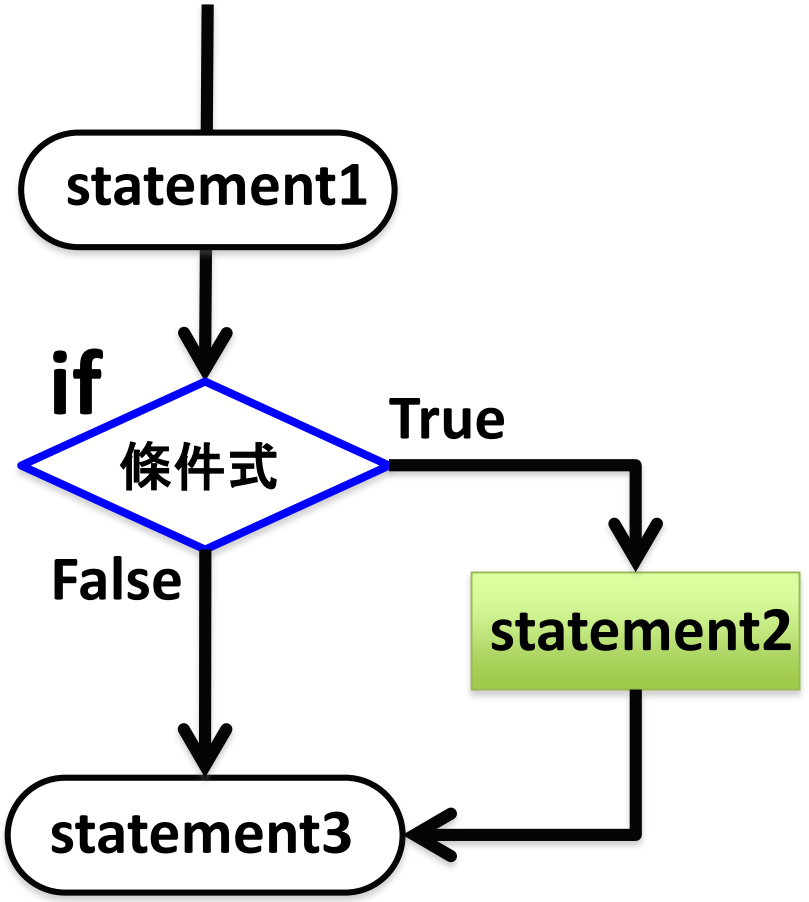
- The evaluated value (1 or 0) will be output on the screen.

5.2 if statement (if 文)

One of the *conditional statement*.

構文(Syntax):

```
statement1;
if (條件式)
    statement2;
statement3;
```

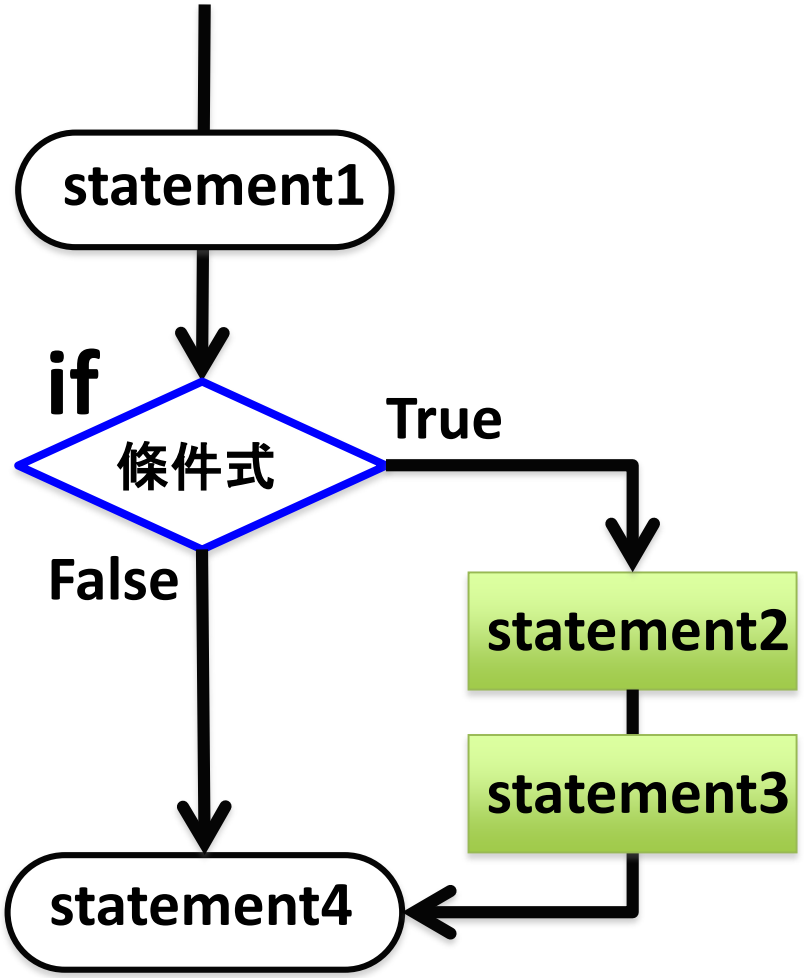


5.2 if statement (if 文)

We can put multiple statements as a block.

構文(Syntax):

```
statement1;
if (條件式){
    statement2;
    statement3;
}

statement4;
```

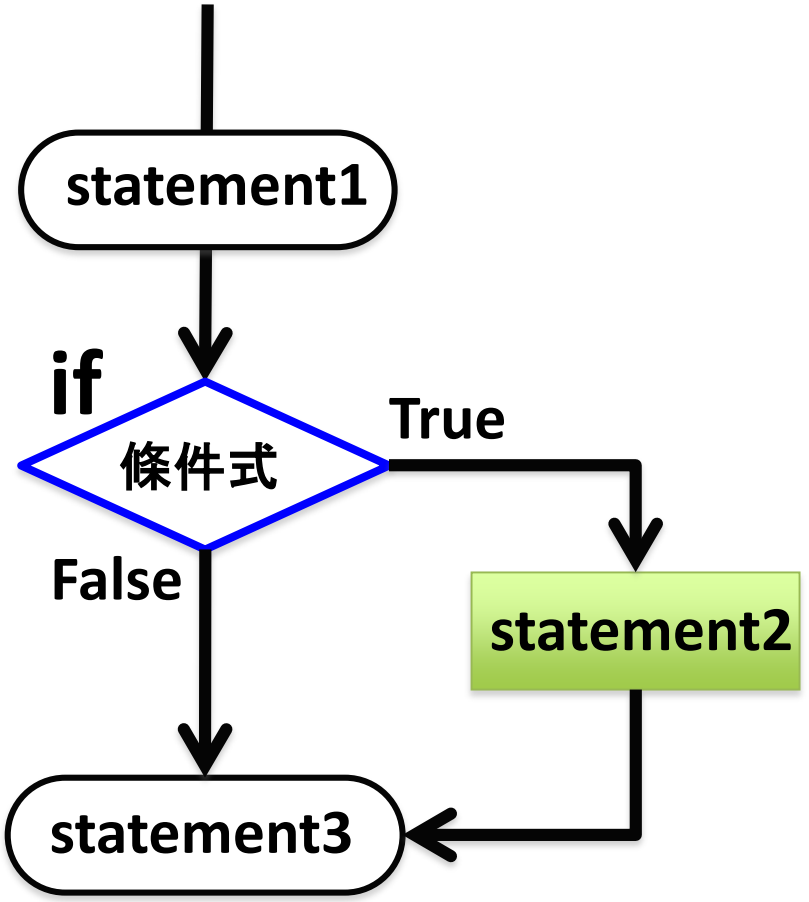


5.2 if statement (if 文)

In order to avoid mistakes, it is better to use {…} even if you put a single statement.

構文(Syntax):

```
statement1;
if (條件式){
    statement2;
}

statement3;
```

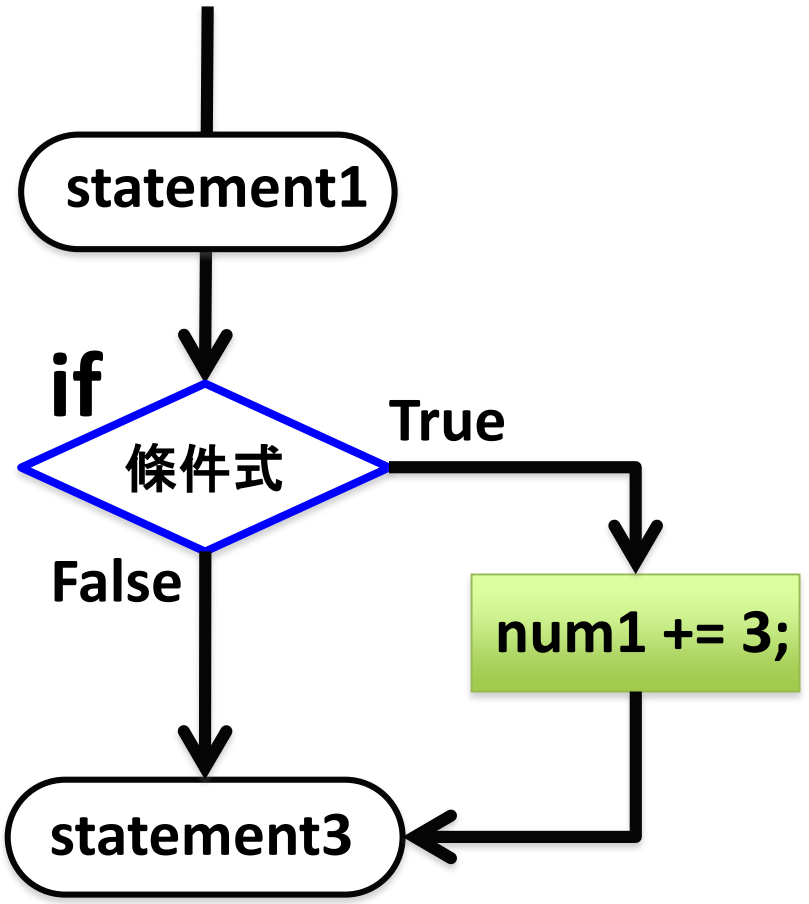


5.2 if statement (if 文)

For a short statement, you can put it in the single line for a clearer appearance of the program.

構文(Syntax):

```
statement1;

if (條件式) num += 3;

statement3;
```

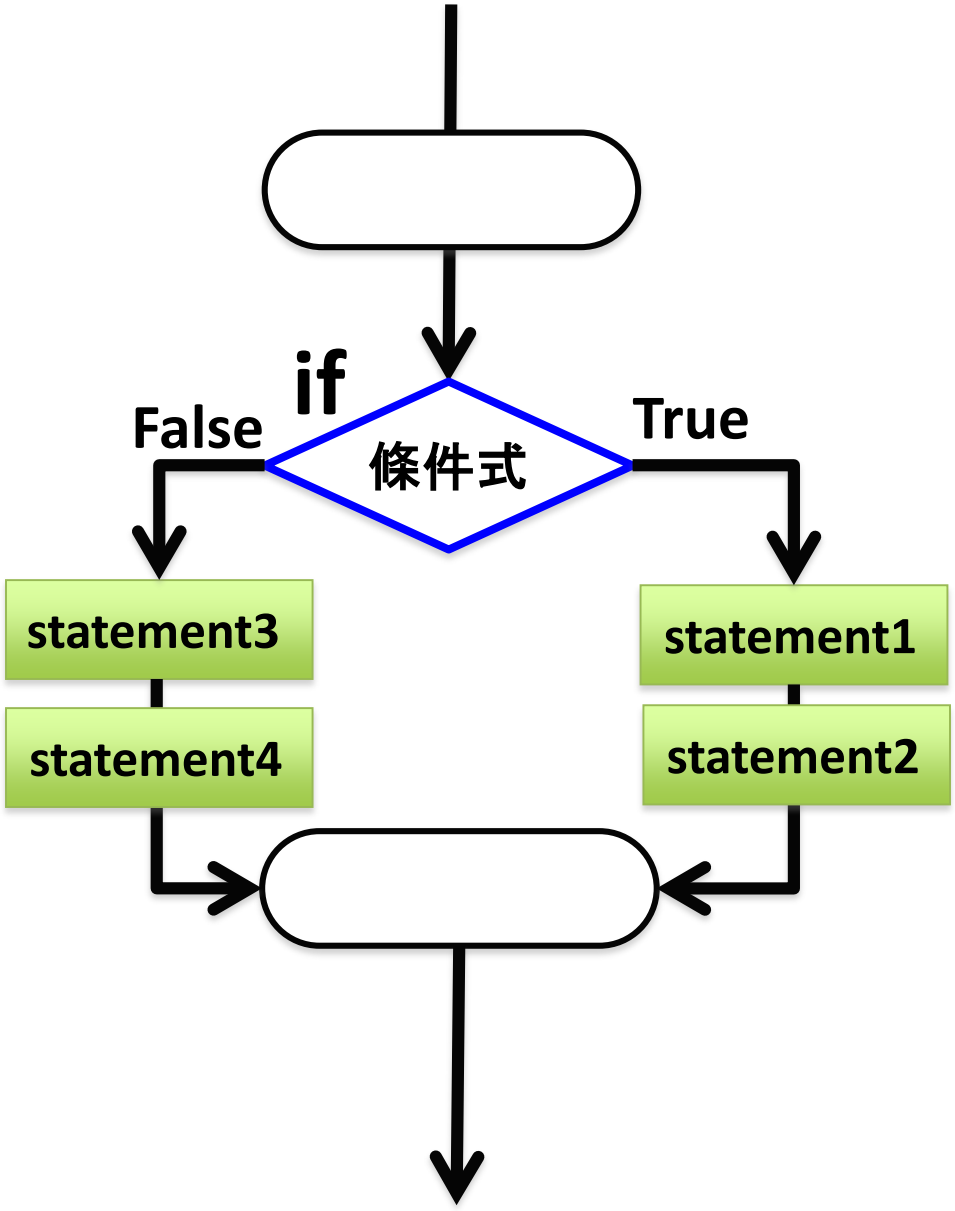


5.3 if... else statement (if...else 文)

if (明天天氣好)
    我去大學。
else
    我在家。

構文(Syntax):

```
if (條件式) {
    statement1;
    statement2;
}//end if
else {
    statement3;
    statement4;
}//end else
```



5.3 if... else statement (if...else 文)

Prepare a new sample file of C codes.

```
#include <stdio.h>

int main (void)
{
	int res;

	printf("Please input an integer. \n");
	scanf("%d", &res);

	if (res <= 5) {
		printf("%d is your input. This is smaller or equal to 5.\n",
res);
	} //end of if
	else {
		printf("%d is your input. This is greater than 5.\n", res);
	} //end of else

	return 0;
}
```

5.4 if… else if… statement (if…else if 文)

Modify the sample file of C codes.

```
#include <stdio.h>

int main (void)
{
	int res;

	printf(“Please input an integer. \n”);
	scanf(“%d”, &res);

	if (res <= 5) {
		printf(“%d is your input. This is smaller or equal to 5.\n”, res);
	} //end of if
	else if (res <= 10) {
		printf(“%d is your input. This is greater than 5 and smaller or equal to 10.\n”,
res);
	} //end of else if
	else {
		printf(“%d is your input. This is greater than 10.\n”, res);
	}
	return 0;
}
```

5.5 switch statement (switch 文)

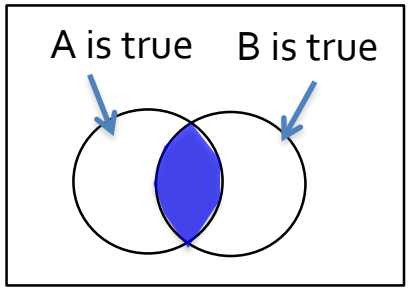
Prepare a new source file of C codes.

```
#include <stdio.h>

int main (void)
{
        int res;
        printf("Please input an integer. \n");
        scanf("%d", &res);
        switch (res) {
                case 1:
                case 2:
                        printf("Your input is 1 or 2.\n");
                        break;
                case 3:
                        printf("Your input is 3.\n");
                        break;
                default:
                        printf("Please input 1, 2 or 3.\n");
                        break;
        } //end of switch
        return 0;
}
```

5.6 Logical operators (邏輯運算子)

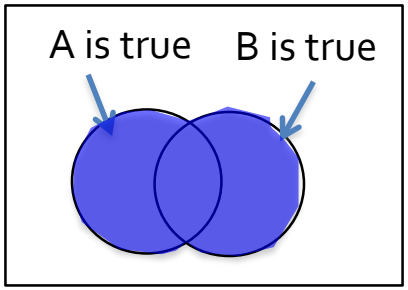
There are 3 logical operators.

A && B



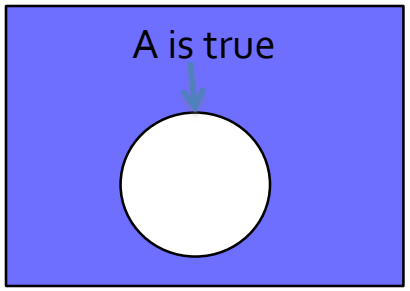
A	B	A &&B
False	False	False
True	False	False
False	True	False
True	True	True

A || B



A	B	A || B
False	False	False
True	False	True
False	True	True
True	True	True

! A



A	!A
False	True
True	False

```
5 > 3 && 3 == 4
a == 6 || a >= 12
! (a == 6)
```

False

True when a = 6 or a >= 12

True when a is not equal to 6

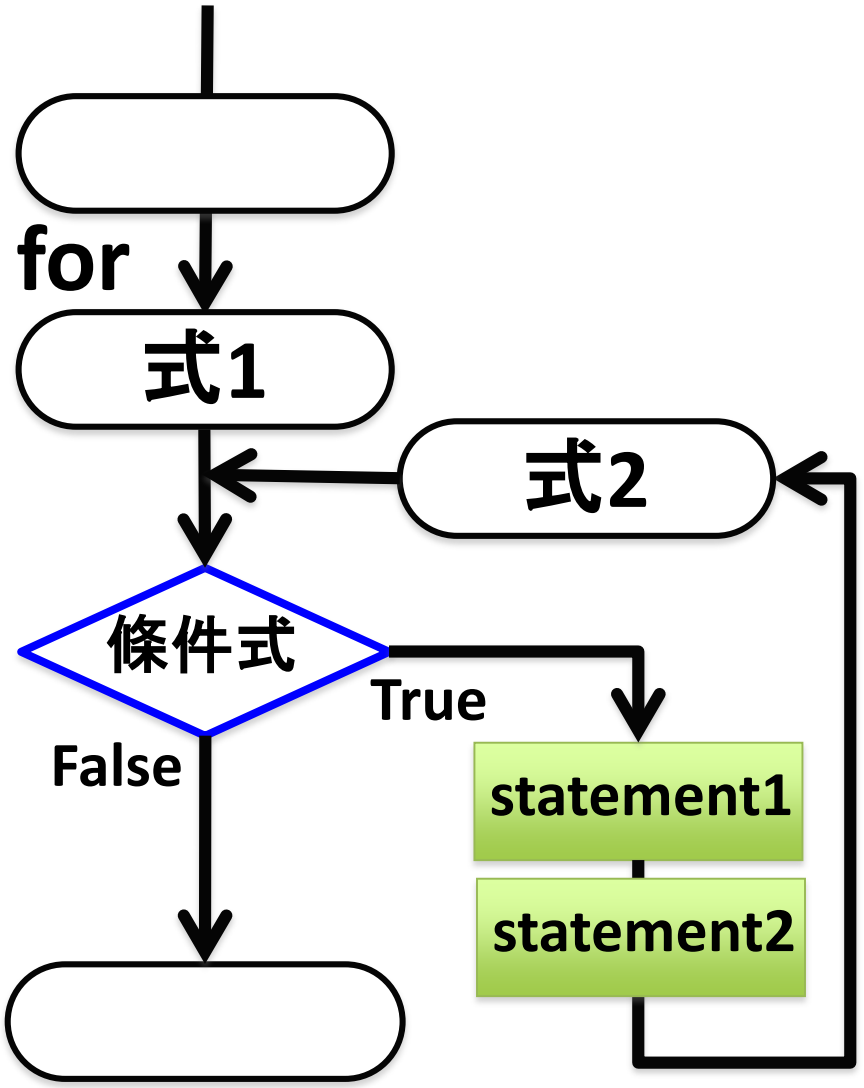
5.7 for statement (for loop)

One of the ***loop statement***.

構文(Syntax):

```
for(式1; 條件式; 式2){
    statement1;
    statement2;
}
```

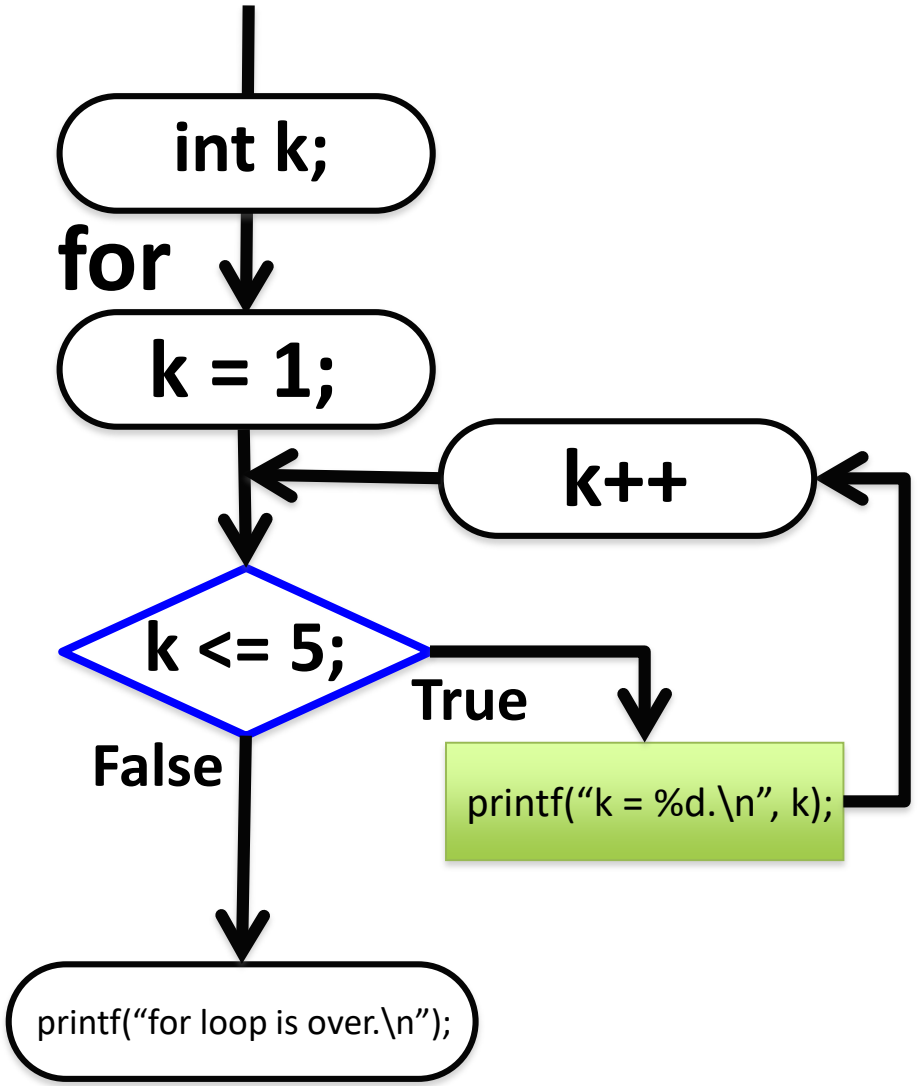
式1→計數器初值: initialization
式2→計數器值更新:renewal



5.7 for statement

Prepare a new source file of C codes.

```
#include <stdio.h>

int main (void)
{
        int k;

        for(k = 1; k <= 5; k++){
                printf("k = %d.\n", k);
        } //end of for k

        printf("for loop is over.\n");

        return 0;
}
```



5.7 for statement

Another example

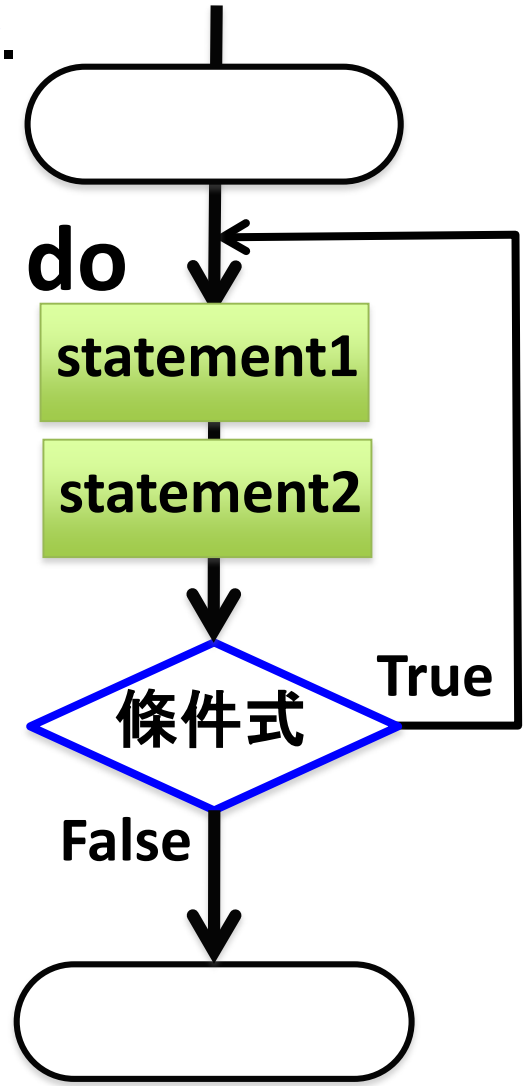
```
#include <stdio.h>

int main (void)
{
        int num = 0;
        int sum = 0;
        int j;

        printf("This program calculates the sum from 1 to input number.\n");
        scanf("%d", & num);

        for(j = 1; j <= num; j++){
                sum += j;
        } //end of for j

        printf("The sum from 1 to %d is %d.\n", num, sum);
        return 0;
}
```

5.8 do…while statement (do…while loop)

Another type of the *loop statement*.

構文(Syntax):

```
do{
    statement1;
    statement2;
} while (條件式)
```



The loop statements are executed at least once.

5.9 Nested structure of statements

You can nest for statements.

```
#include <stdio.h>

int main (void)
{
        int j, k;
        int product;
        for(j = 0; j < 5; j++) {
                for(k = 0; k < 3; k++) {
                        product = j*k;
                        printf("(j, k) = (%d, %d).\t j*k = %d.\n", j, k, product);
                }//end of for k
        }//end of for j

        return 0;
}
```

Nested loop

5.10 break and continue statement

You can change the flow of program with a break statement.

```
#include <stdio.h>

int main (void)
{
        int j, res;

        printf("When do you like to stop the loop? (1-10)\n");
        scanf("%d", &res);

        for(j = 1; j<= 10; j++) {
                printf("j = %d.\n", j);
                if(j == res) {
                        printf("The loop is interrupted.\n");
                        break;
                }//end of if
         }//end of for j
         return 0;
}
```

5.10 break and continue statement

You can change the flow of program with a continue statement.

```
#include <stdio.h>

int main (void)
{
      int j, res;

      printf("When do you like to skip the loop? (1-10)\n");
      scanf("%d", &res);

      for(j = 1; j<= 10; j++) {
            if(j == res) {
                  printf("The treatment is skipped at j = %d.\n", j);
                  continue;
             }//end of if
            printf("j = %d.\n", j);
}//end of for j
       return 0;
}
```

Homework this week

(1) Write a source code for the program that can judge if the input is even or odd number.

Please input an integer number.	(*output*)
3 (or 4) ⏎	(*input*)
3 (or 4) is an odd (or even) number .	*(output)*

(2) Write a source code for the program that can judge you are overweight or underweight compared to the standard weight.

Please input your height (cm) and weight (kg).	(*output*)
172.3 ⏎	(*input*)
65.0 ⏎	(*input*)
Your BMI is 21.890.	(*output*)
The standard BMI is assumed to be 24.0.	(*output*)
I'm afraid you are overweight (or underweight).	(*output*)

HINT: BMI = (weight kg)/(height m)^2, and standard BMI is assumed here to be 24.0.

Homework this week

(3) Write a source code for the program that can judge if the input number is a prime number (質數) or not.

Please input an integer number (> 1). (*output*)
7 ⏎ (*input*)
7 is a prime number. *(output)*

Please input an integer number (> 1). (*output*)
15 ⏎ (*input*)
15 is not a prime number. *(output)*