Practice Information Processing

（IMACU）

Second lecture (part 2)：
Summary of Lectures/Basic of Program

Makoto Hirota

From human thinking to computer programs

- How to make your computer handle a problem?
  - Humans can describe the flow of processing in English.

    Example) Grade evaluation method

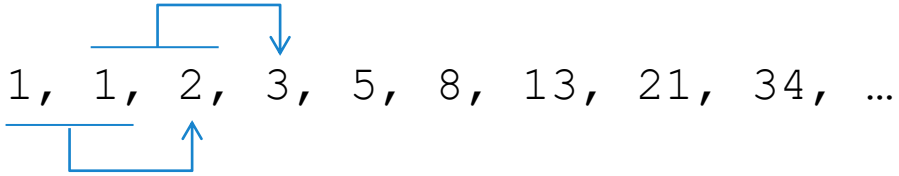
    - Emphasis on attendance. I would like to give a passing score to the students who attended each time.
    - I will add points to your voluntary efforts. In order to take AA, it is necessary to tackle voluntary challenges.

- How to write a problem with a computer program?

Describe
<Using mathematical formulas>
<Processing flow>

Mathematical expressions and program examples

- The Fibonacci sequence is a sequence in which the previous two numbers are added to obtain the next number. Here, the first and second numbers are both 1. Find the nth Fibonacci number



<Formula> (iterative processing)

Initial condition

$$F_1 = 1$$
$$F_2 = 1$$

Repeats

first: $F_3 = F_1 + F_2$

second: $F_4 = F_2 + F_3$

⋮ ⋮

(n-2)th: $F_n = F_{n-2} + F_{n-1}$

<program example> fibo1.c

```
/**** variable declaration ****/
int f1, f2, f3;
int i, n;

 /**** processing contents****/
f1 = 1;
f2 = 1;
n = 8;

for(i=0;i < n-2; i++){
    f3 = f1 + f2;
    f1 = f2;
    f2 = f3;
}
printf("n=%d: fn=%d¥n", n, f3);
```

Initial condition

(the case of n=8)

repeating（(n-2)times）

Examples of Fibonacci sequences (2) (3)

<Math>（recursive expression）

$$F_n = F_{n-2} + F_{n-1}$$

where,

$$F_1 = 1, F_2 = 1$$

Recursive expression: A reference to the description itself appears, when describing something. (Here, "F" is used to define "F")

<program example>

fibo2.c

```
int fibonacci(int n)
{
    int fn;

    if (n == 1 || n == 2){

        fn = 1;

    }else{

        fn = fibonacci(n-2) + fibonacci(n-1);

    }

    return fn;
}
```

functionalize F

recursive call
（Calling itself in the function）

※recursive expression will be explained in the following lectures

<Math>(Expression by general term)

The general term of the Fibonacci number is expressed by the following equation

$$F_n = \frac{1}{\sqrt{5}}\left\{\left(\frac{1+\sqrt{5}}{2}\right)^n - \left(\frac{1-\sqrt{5}}{2}\right)^n\right\}$$

<program example>

fibo3.c

```
/**** variable declaration****/
int fn, n;

 /**** processing contents****/
n = 8;
fn = round((pow((1+sqrt(5))/2, n)
        - pow((1-sqrt(5))/2, n))/sqrt(5));

printf("n=%d: fn=%d¥n", n, fn);
```

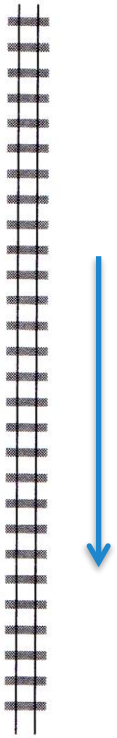
Use mathematical functions

`round()`: Round to the nearest whole number
`sqrt()`: Square root
`pow(a, n)`: Exponentiation （a^n）

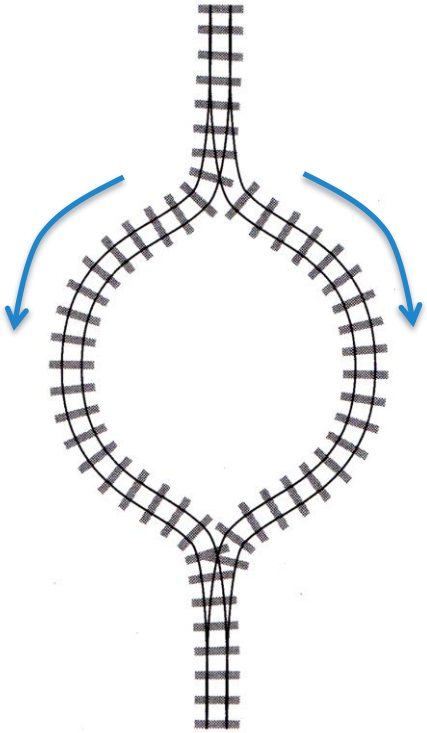
※-lm option is required for compilation

Basic structure of processing flow

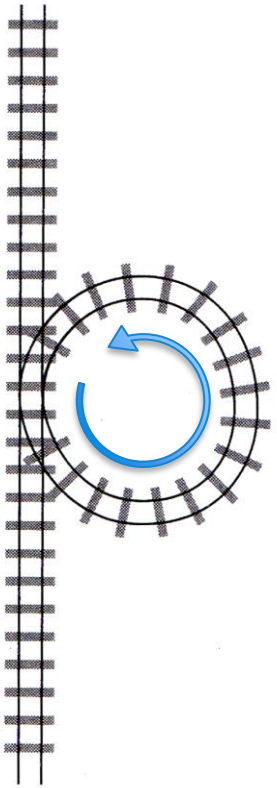
- There are only three basic forms of "processing flow" in a program



Sequence



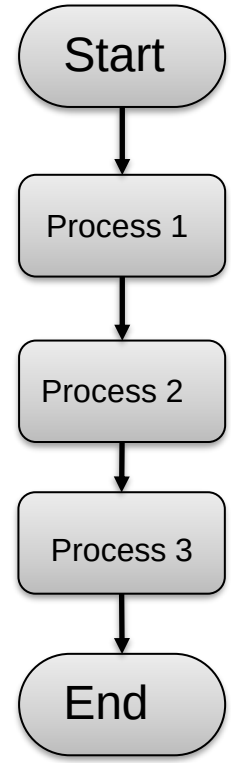
Selection



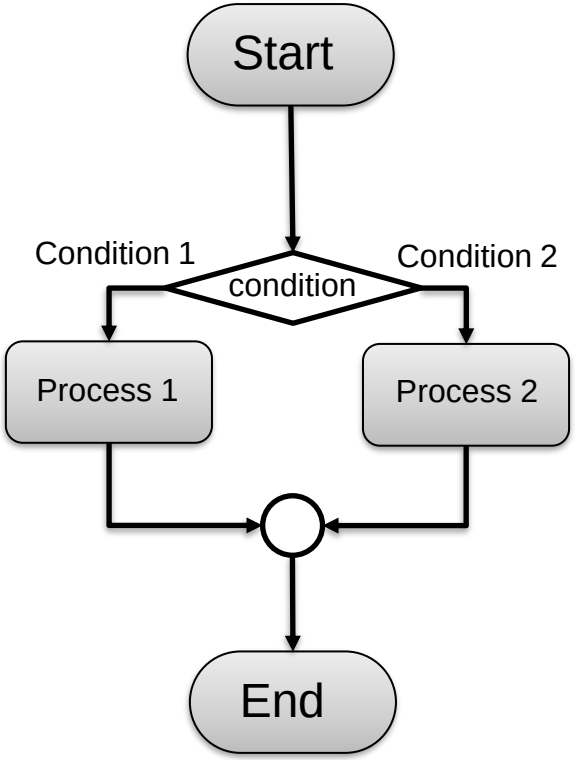
Iteration

Flowchart: processing flow

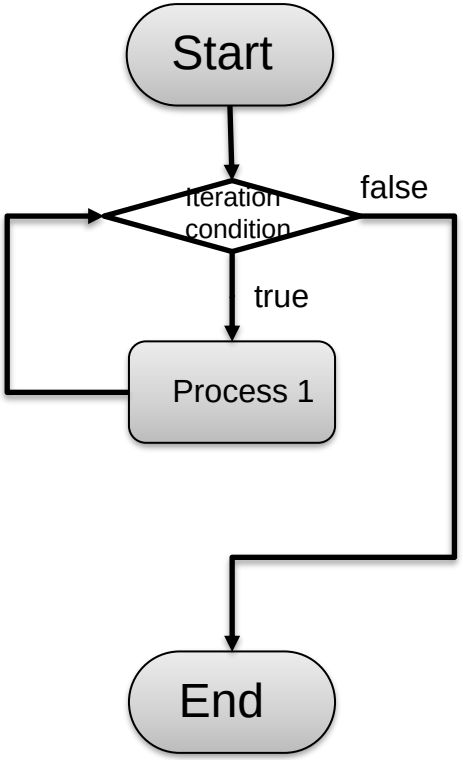
- Expression using flowchart



Sequence

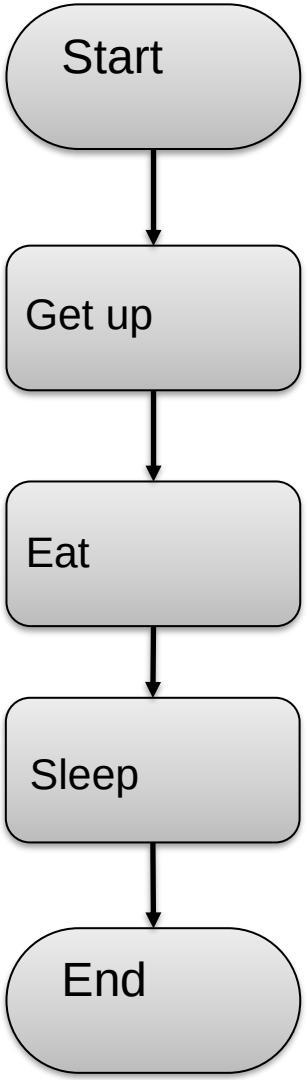


Selection



Iteration

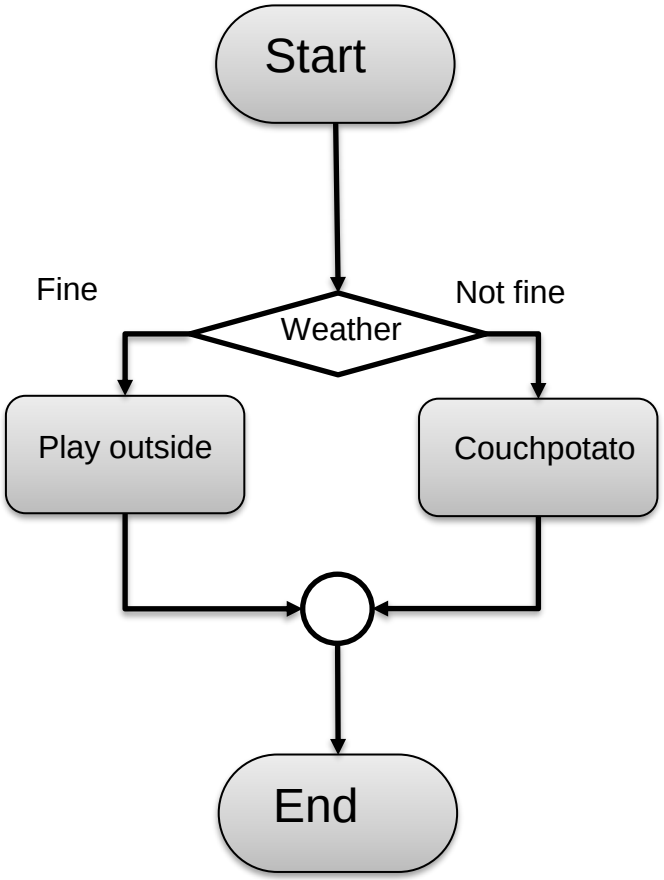
Sequence process



C-language-like description

```
Get up;
Eat;
Sleep;
```

Selection process（`if` statement）



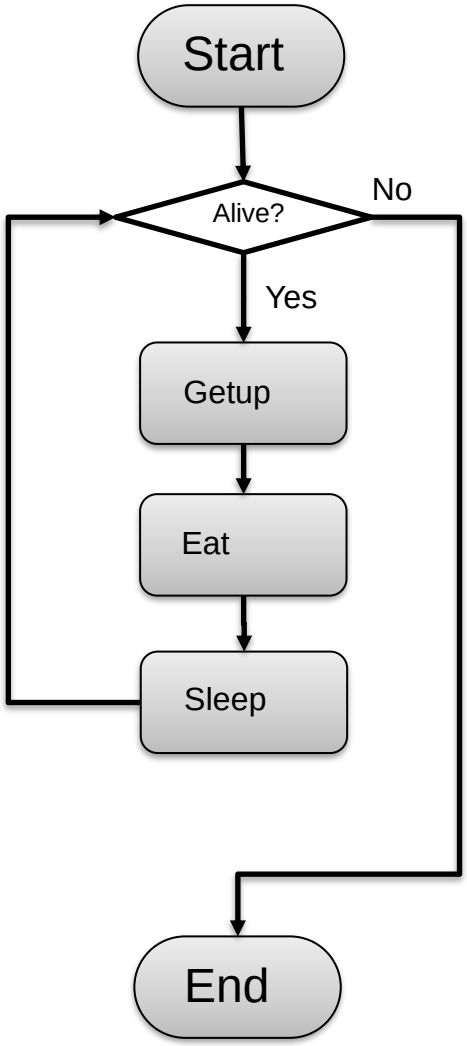
Selection

C-language-like description

```
if (weather == fine){
        play outside;
} else {
        couchpotato;
}
```

※next lecture

Iteration process（while statement）



Iteration

C-language-like description

```
while (alive){
        get up;
        eat;
        sleep;
}
```

※next lecture

Issues in flowchart

- Flowchart for complex processing is confusing

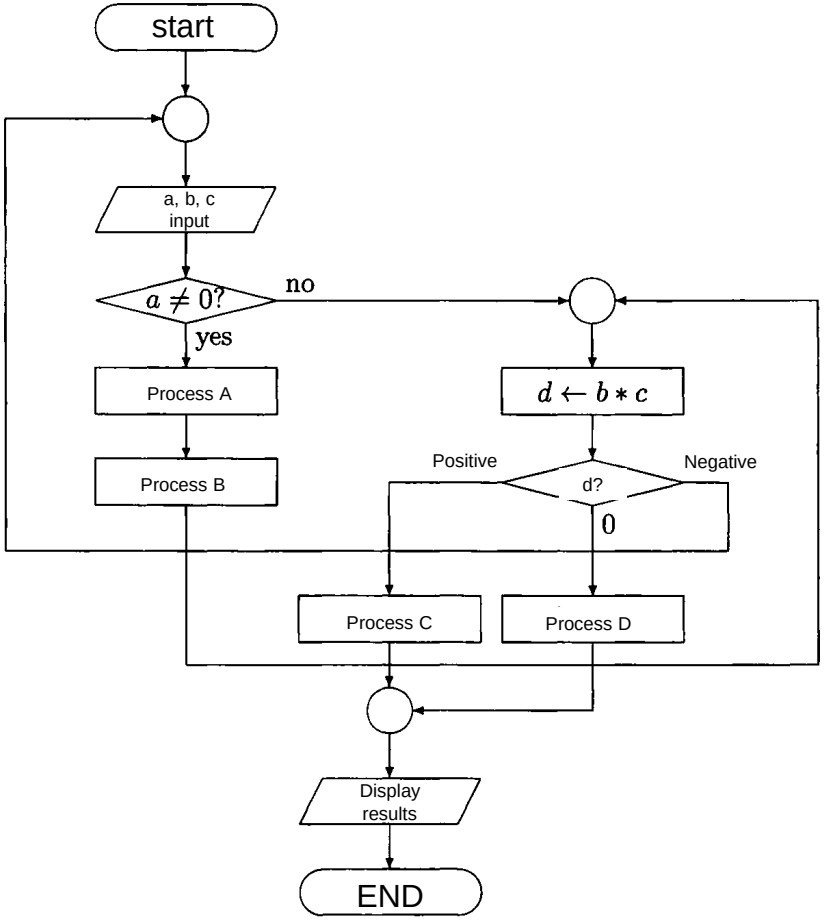
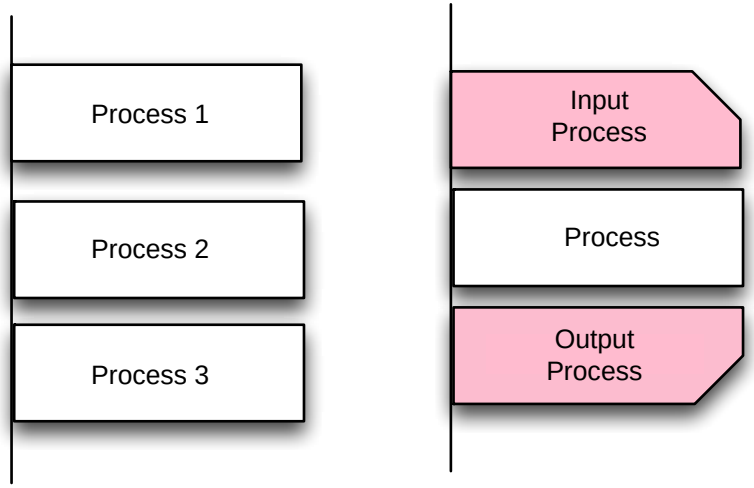


Figure indicating confusing flowchart

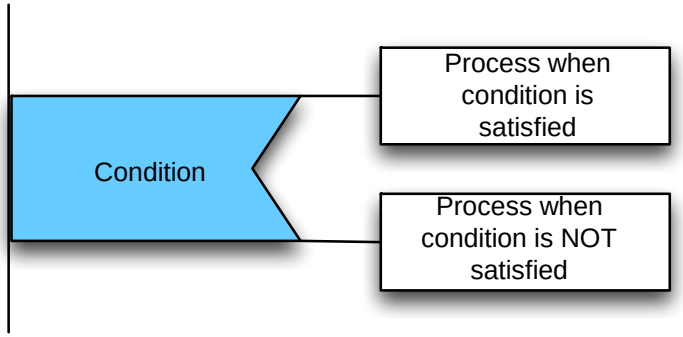
Expression using PAD（Problem Analysis Diagram）



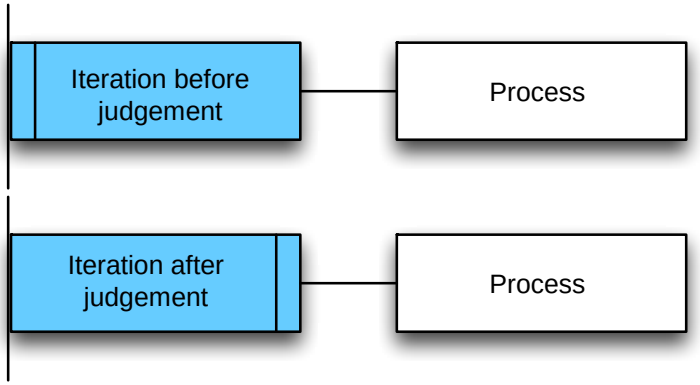
Sequence

Process in order of
Top to bottom
Left to right

Details in next week



Selection



Iteration

Iteration of a certain number of times

- `for` statement

```
for （Initial condition ； continuation condition； Incremental process）
{
    Repeating process ；
}
```

- Execution order of `for` statement

1. Execute the expression of the initial condition
2. Execute the processing in the `for` statement block
3. Perform incremental processing
4. If the continuation condition is true, return to 2.

Exercise 2-3: `for` statement

<loop.c>

```c
#include <stdio.h>

int main()
    /**** variable declaration ****/
    int i;

     /**** processing contents****/
    for(i=0;i < 10; i++){
        printf("hello: %d¥n", i);
    }

    return 0;
}
```

- Write the following program using templete.c and check the execution result.
- Filename : loop.c
- Compile:

```
$ gcc -Wall -o loop loop.c
$ ./loop
```

Output results


```
$ ./loop
hello: 0
hello: 1
hello: 2
hello: 3
hello: 4
hello: 5
hello: 6
hello: 7
hello: 8
hello: 9
```

Why the numbers are from 0 to 9
Please check by looking at the program

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

}
```

- Here, the notation "i ++" is the same as i=i+1.

Exercise 2-4: Practice of `for` statement

- Modify the previous loop.c to create a program that performs the following calculation.
  - loop01.c
    - A program that calculates the sum from 1 to 100 and displays the result
  - loop02.c
    - A program that calculates the sum of numbers that are multiples of 3 from 1 to 1000 and displays the result.

Exercise 2-5: Calculation of factorial：factorial.c

- Please create a program to output factorial !n for the natural number n entered from the keyboard

Definition of factorial

$$n! = n \times (n-1) \times (n-2) \times \cdots \times 3 \times 2 \times 1$$

Summary

- C language basics
  - Variables and data types
  - How to use the `printf` function
- From human thinking to computer programs
  - Three basic forms of processing flow
  - “Sequence”, “Selection”, “Iteration”
  - Flowchart and PAD representation
- Basic form “Iteration of a certain number of times”
  - How to write a `for` statement

Lecture next week

- Example of model answer (`for` statement)
- Selection format
  - `if` statement
    - `if`
    - `if else`
    - `else`
  - `switch` Statement
- Iteration format No. 2
  - `while` statement
- Array(1)