



www.siam2dev.com

วิชา การเขียนโปรแกรมเชิงวัตถุ (4122309)

บทที่ 1 หลักการเขียนโปรแกรมเบื้องต้น (Fundamental of Programming)



ผู้ช่วยศาสตราจารย์ ดร. นัฐพงศ์ ส่องเนียม

Asst. Dr. Nattapong Songneam

เว็บไซต์ :: <http://www.siam2dev.com>

อีเมล :: siam2dev@gmail.com

สาขาวิชาวิทยาการคอมพิวเตอร์

คณะวิทยาศาสตร์และเทคโนโลยี มหาวิทยาลัยราชภัฏพระนคร

Last Update : 17/07/2566



บทที่ 1 หลักการเขียนโปรแกรมเบื้องต้น

(Fundamental of Programming)

Agenda

1. คำอธิบายรายวิชา
2. ทบทวนเนื้อหาการเขียนโปรแกรมเบื้องต้น
 1. ประวัติโปรแกรมเมอร์คนแรกของโลก
 2. หลักการทำงานของภาษาคอมพิวเตอร์
 3. ประเภทของตัวแปลภาษา
 4. ระดับของภาษาคอมพิวเตอร์
 5. ขั้นตอนการเขียนโปรแกรมคอมพิวเตอร์
 6. การวิเคราะห์ปัญหา
 7. ผังงาน
3. หลักการเชิงวัตถุเบื้องต้น

คำอธิบายรายวิชา

การพัฒนาระบบงานทางคอมพิวเตอร์ด้วยวิธีการของเทคโนโลยีเชิงวัตถุ การสร้างชนิดข้อมูล คลาส คุณสมบัติของวัตถุ การรับทอด การห่อหุ้ม โพลีมอร์ฟิซึม การวิเคราะห์และการออกแบบระบบงานเชิงวัตถุ API (Application Programming Interface) ของภาษาที่ใช้ในการเขียนโปรแกรมเชิงวัตถุ และการพัฒนาโครงการงานเชิงวัตถุ

Augusta Lovelace Ada คือ โปรแกรมเมอร์คนแรกของโลก แต่ Edsger Wybe Dijkstra ใช้คำว่า โปรแกรมเมอร์ (Programmer) กับโลกของคอมพิวเตอร์เป็นคนแรก



ADA

ข้อมูลจาก <http://www.twoguru.com/playground/article/dijkstra.htm>

ข้อมูลจาก <http://www.cs.utexas.edu/users/dahlin/Classes/GradOS/papers/p341-dijkstra.pdf>

ข้อมูลจาก <http://www.twoguru.com/playground/article/dijkstra.htm>

ข้อมูลจาก <http://www.thaiall.com/article/teachpro.htm>



อดา ไบรอน (พ.ศ. 2358-2395)
ಅದಾ ಲೂವಲೇ (ಮ.ವ. 5328-5362)

- "what if a calculating engine could not only foresee but could act on that foresight"
- (จะเป็นอย่างไร ถ้าหากเครื่องคำนวณไม่เพียงสามารถหยั่งรู้ได้ หากแต่สามารถตอบสนองต่อการหยั่งรู้นั้นได้ด้วย)
- สิ่ง que เธอทำคือ การสร้างภาษาสำหรับเครื่องวิเคราะห์ (analytical engine)



เอดา ไบรอน (พ.ศ. 2358-2395)

- แต่งเพลงที่ซับซ้อน
 - สร้างภาพกราฟิก
 - นำมาใช้เพื่อการคำนวณขั้นสูง
 - พัฒนาวงการวิทยาศาสตร์ได้
-
- ร้อยกว่าปีต่อมา ในปี พ.ศ. 2522 (ค.ศ. 1979) กระทรวงกลาโหมสหรัฐฯ สร้างภาษาคอมพิวเตอร์มาตรฐาน ISO ขึ้นมาตัวแรก พร้อมตั้งชื่อเพื่อเป็นเกียรติแก่ เลดี้ เอดา ว่า ภาษา "ADA"

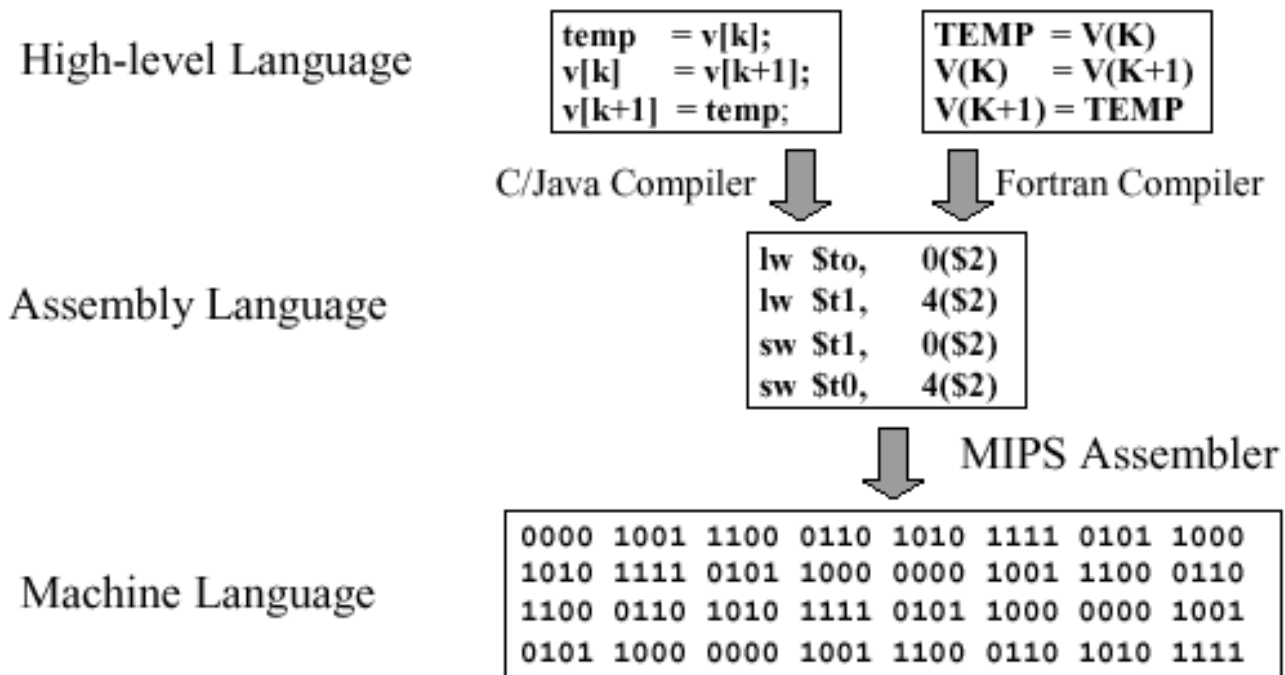


Figure 2.2. Basic ISA cycle, where fetch gets operands from memory, decode translates the operands into a format the ALU can accept, and execute performs an ALU operation on the decoded operands - adapted from [Maf01]

ที่มา : <https://www.cise.ufl.edu/~mssz/CompOrg/CDA-lang.html>

2.3.2 Machine Language

0 → ปิด
1 → เปิด

ENCODING

Bit บิต

ASCII CODE 8 bit : Binary Digit

00000000
00000001
00000010
...
11111111

UNI CODE 16 bit : Binary Digit

$$2^{16} = 32768$$

$$2^8 = 256$$

2.3.3 ASCII

- 8 Bit = 1 Byte
 - 01010101 = 256 = 2^8
 - 256 แบบ 44 26
 - 1024 Byte = 1 Kilo
 - 1MB = 1024 k = $1024 * 1024$ byte
 - 1GB = 1024MB = $1024 * 1024 * 1024$
 - 1 TB = 1024GB
- 2MB
8 MB
16 MB
32 MB
8 GB

2.3.3 Unicode Text

ยูนิโคด คือ รหัสคอมพิวเตอร์ใช้แทนตัวอักษรสามารถใช้แทน ตัวอักษร,ตัวเลข,สัญลักษณ์ต่างๆ ได้มากกว่ารหัสแบบเก่าอย่าง ASCII ซึ่งเก็บตัวอักษรได้สูงสุดเพียง 256 ตัว(รูปแบบ) โดย Unicode รุ่นปัจจุบันสามารถเก็บตัวอักษรได้ถึง 34,168 ตัวจากภาษาทั้งหมดทั่วโลก 24 ภาษา โดยไม่สนใจว่าเป็นแพลตฟอร์มใด ไม่ขึ้นกับโปรแกรมใด หรือภาษาใด unicode ได้ถูกนำไปใช้โดยผู้นำในอุตสาหกรรม เช่น Apple, HP, IBM, Microsoft, Unix ฯลฯ และเป็นแนวทางอย่างเป็นทางการในการทำ [ISO /IEC 10646](#) ดังนั้น Unicode จึงถือเป็นมาตรฐานในการกำหนดรหัส สำหรับทุกตัวอักษร ทุกอักษร unicode ทำให้ข้อมูลสามารถเคลื่อนย้ายไปมาในหลายๆ ระบบ ข้ามแพลตฟอร์มไปมา หรือข้ามโปรแกรมได้อย่างสะดวก โดยไร้ข้อจำกัด

Unicodeต่างจาก ASCII คือ ASCII เก็บ byte เดียว แต่ Unicode เก็บ 2 byte ซึ่งข้อมูล 2 byte เก็บข้อมูลได้มากมายมหาศาล สามารถเก็บข้อมูลได้มากมายหลายภาษาในโลกอย่างภาษาไทยก็อยู่ใน Unicode นี้ด้วยเหมือนกัน ดังนั้นรหัสภาษาไทยเอาไปเปิดในภาษาจีน ก็ยังเป็นภาษาไทยอยู่ ไม่ออกมาเป็นภาษาจีน เพราะว่ามี code ตายตัวอยู่ว่า code นี้จงไว้สำหรับภาษาไทย แล้ว code ตรงช่วงนั้นเป็นภาษาจีน ตรงโน้นเป็นภาษาญี่ปุ่น จะไม่ใช้ที่ซ้ำกัน เป็นต้น

ที่มา: <https://www.mindphp.com/คู่มือ/73-คืออะไร/2050-unicode-คือ-อะไร.html>

Unicode Text

A	0000	0000	0100	0001
S	0000	0000	0101	0011
C	0000	0000	0100	0011
I	0000	0000	0100	1001
I	0000	0000	0100	1001
	0000	0000	0010	0000
天	0101	1001	0010	1001
地	0101	0111	0011	0000
	0000	0000	0010	0000
س	0000	0110	0011	0011
ل	0000	0110	0100	0100
ا	0000	0110	0010	0111
م	0000	0110	0100	0101
	0000	0000	0010	0000
a	0000	0011	1011	0001
¢	0010	0010	0111	0000
γ	0000	0011	1011	0011

ภาษาคอมไพเตอร์ หมายถึง

- หมายถึง สัญลักษณ์ที่ผู้คิดพัฒนาภาษากำหนดขึ้นมา
- เพื่อใช้แทนคำสั่งสื่อสารสั่งงาน ระหว่างมนุษย์กับเครื่องคอมไพเตอร์และอุปกรณ์ต่อพ่วงอื่น ๆ

ภาษาคอมไพเตอร์ หมายถึงภาษาที่ใช้ในการโต้ตอบและสื่อสารกับคอมไพเตอร์ ภาษาเหล่านี้มีหลายแบบและมีวัตถุประสงค์ในการใช้งานต่างกัน ภาษาคอมไพเตอร์สามารถใช้ในการเขียนโปรแกรมและสร้างแอปพลิเคชัน โดยภาษาคอมไพเตอร์ที่ได้รับความนิยมและถูกนำมาใช้ในการพัฒนาซอฟต์แวร์หรือโปรแกรมอย่างกว้างขวางได้แก่ C++, Java, Python, JavaScript, Ruby, PHP, Swift, Objective-C, C#, Visual Basic, และอื่นๆ ภาษาคอมไพเตอร์มีไวยากรณ์และกฎในการเขียนเช่นเดียวกับภาษามนุษย์ แต่จะถูกออกแบบมาเพื่อให้คอมไพเตอร์สามารถเข้าใจและประมวลผลได้ในลักษณะที่ต้องการ

ภาษาคอมพิวเตอร์มีมากมาย

ข้อมูลจาก <http://dmoz.org/Computers/Programming/Languages/> 143 ภาษา
ข้อมูลจาก <http://people.ku.edu/~nkinners/LangList/Extras/langlist.htm> 2500 ภาษา
ข้อมูลจาก <http://www.levenez.com/lang/> 50 ภาษา (Computer Languages Timeline)

ABC, Ada, ADL, Algol 60, Algol 68, APL, AppleScript@,
Assembly, Awk, BASIC, Befunge, BETA, Bigwig, Bistro@,
Blue, Brainfuck, C, C++, Caml@, Cecil, CHILL, Clarion,
Clean, Clipper, CLU, Cobol, CobolScript, Cocoa, Component
Pascal@, C-sharp, Curl, D, DATABUS, Delphi, DOS Batch@,
Dylan, E, Eiffel, Elastic, Erlang, Euphoria, Forth,
Fortran, Fortress, FP, Frontier, Goedel, Groovy@, Haskell,
HTML@, HTMLScript@, HyperCard@, ICI, Icon, IDL, Intercal,
Io, Jal@, Java, JavaScript, Jovial, LabVIEW, Lagoon@,
LaTeX@, Leda, Limbo, Lisp, Logo@, Lua, m4, Maple@,
Mathematica@, MATLAB@, Mercury, Miranda, Miva, ML, Modula-
2, Modula-3, Moto, Oberon, Objective Caml@, Objective-C,
Obliq, Occam, Oz, Pascal, Perl, PHP, Pike, PL, Pliant, PL-
SQL, POP-11, PostScript@, PowerBuilder@, Prograph, Prolog,
Proteus, Python, R@, REBOL, Refal, Rexx, Rigal, RPG, Ruby,
SAS, Sather, Scheme@, Self@, SETL, SGML@, Simkin, Simula,
Sisal, S-Lang, Smalltalk, Snobol, SQL, Squeak@, Tcl-Tk,
Tempo, TeX@, TOM, TRAC, Transcript, Turing, T3X, UML@,
VBScript@, Verilog@, VHDL@, Visual Basic, Visual
DialogScript, Visual FoxPro, Water, XML@, XOTcl@, YAFL,
Yorick, Z

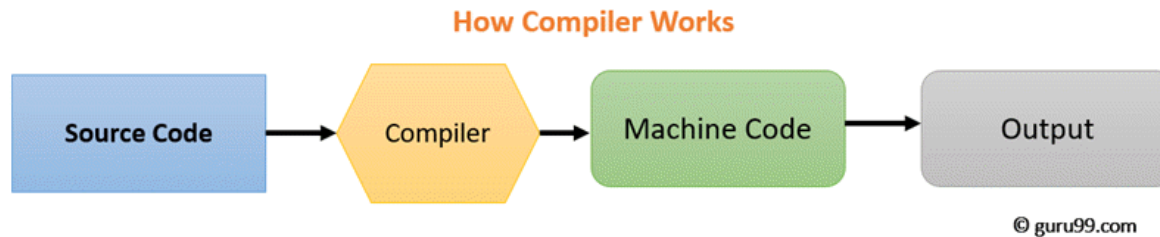
ส่วนประกอบของภาษาคอมพิวเตอร์

- Source Code
- Compiler ตัวแปล ภาษา...
 - Compiler
 - Interpreter
- ไวยากรณ์ทางภาษา Syntax



ตัวแปลภาษาแบ่งออกเป็น 2 ชนิด ดังนี้

- Interpreter แปลคำสั่งทีละบรรทัดแล้วทำงาน ตัวอย่าง ภาษา Basic
- Compiler อ่านรหัสต้นฉบับมาทั้งหมดแล้วแปลครั้งเดียว ตัวอย่าง ภาษา ปาสคาล ภาษาซี ซีพลัสพลัส เป็นต้น



ที่มา:

<https://www.guru99.com/difference-compiler-vs-interpreter.html>

1. ภาษาเครื่อง (Machine Language)
2. ภาษาแอสเซมบลี (Assembly)
3. ภาษาระดับสูง (High-Level Language 4gl) ได้แก่ ภาษาซี ภาษาปาสคาล ภาษา C++ , java , VB, Delphi ฯลฯ
4. ภาษาระดับสูงมาก (Very High-Level Language 5gl) ภาษานี้ถูกออกแบบมาเพื่อให้ใกล้เคียงกับภาษามนุษย์มากยิ่งขึ้น โดยเราบอกเพียงความต้องการแล้วตัวภาษาจะทำงานหลายอย่างแทน ตัวอย่าง SQL (อ่านว่า ซีควอล)
5. ภาษาธรรมชาติ (Natural Language) มีการใช้ A.I. ปัญญาประดิษฐ์ เพื่อให้สามารถคิด หรือรับรู้ ได้

- Prolog
- Lisp

AI
Neural Network
Fuzzy

2.5.1 ภาษาเครื่อง (Machine language)

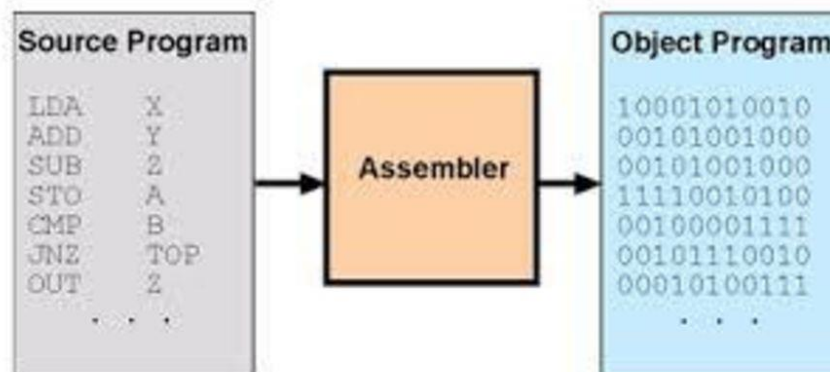
{00P}

- เป็นภาษาพื้นฐานที่คอมพิวเตอร์สามารถเข้าใจได้ แต่ละคำสั่งประกอบขึ้นจากกลุ่มตัวเลข 0 และ 1 ซึ่งเป็นเลขฐานสอง

แอสเซมบลี	ภาษาเครื่อง
LD A,(8000)	00111010,00000000,10000000
LD B,A	01000111
LD A,(8001)	00111010,00000001,10000000
ADD A,B	10000000
LD (8002),A	00110010,00000010,10000000

- เป็นภาษาที่ใช้สัญลักษณ์ข้อความแทนกลุ่มของตัวเลขฐานสอง เพื่อให้ง่ายต่อการเขียนและการจดจำมากขึ้น การทำงานของโปรแกรมจะต้องทำการแปลภาษาแอสเซมบลีให้เป็นภาษาเครื่อง โดยใช้ตัวแปลที่เรียกว่า แอสเซมเบลอร์ (Assembler)

แอสเซมเบลอร์ (Assembler)



ภาษาแอสเซมบลี (อังกฤษ: Assembly Language) หมายถึง ภาษาที่ใช้ในการเขียนโปรแกรมภาษาหนึ่งซึ่งจะทำงานโดยขึ้นกับรุ่นของไมโครโพรเซสเซอร์ หรือ "หน่วยประมวลผล" (CPU) ของเครื่องคอมพิวเตอร์ การใช้ภาษาแอสเซมบลีจำเป็นต้องผ่านการแปลภาษาด้วยคอมไพเลอร์เฉพาะเรียกว่า แอสเซมเบลอร์ (assembler) ให้อยู่ในรูปของรหัสคำสั่งก่อน (เช่น .OBJ) โดยปกติ ภาษานี้ค่อนข้างมีความยุ่งยากในการใช้งาน และการเขียนโปรแกรมเป็นจำนวนบรรทัดมากกว่า เมื่อเปรียบเทียบกับการใช้ภาษาระดับสูง เช่น ภาษา C หรือภาษา BASIC แต่จะทำให้ได้ผลลัพธ์การทำงานของโปรแกรมเร็วกว่า และขนาดของตัวโปรแกรมมีขนาดเนื้อที่น้อยกว่าโปรแกรมที่สร้างจากภาษาอื่นมาก จึงนิยมใช้ภาษานี้เมื่อต้องการประหยัดเวลาทำงานของเครื่องคอมพิวเตอร์ และเพิ่มประสิทธิภาพของโปรแกรม เนื่องจากตัวคำสั่งภายในภาษาอ้างอิงเฉพาะกับรุ่นของหน่วยประมวลผล ดังนั้นถ้ามีการเปลี่ยนแปลงไปใช้กับหน่วยประมวลผลอื่นหรือระบบอื่น (เช่น หน่วยประมวลผล x86 ไม่เหมือนกับ z80) จะต้องมีการปรับแก้ตัวคำสั่งภายในซึ่งบางครั้งอาจไม่สามารถปรับปรุงแก้ไขได้อย่างสมบูรณ์ (วิกิพีเดีย)

แอสเซมบลี	ภาษาเครื่อง
LDA,(8000)	00111010,00000000,10000000
LD B,A	01000111
LDA,(8001)	00111010,00000001,10000000
ADDA,B	10000000
LD (8002)A	00110010,00000010,10000000

ตัวอย่างโปรแกรมภาษา แอสเซมบลี

```
; This program prints the message "Hello world"
```

```
dseg segment
    msg1 db 'Hello world',10h,13h,'$'
dseg ends
```

```
sseg segment stack
    db 100 dup (?)
sseg ends
```

```
cseg segment
    assume cs:cseg,ds:dseg,ss:sseg
start:
    mov ax,dseg ;set DS
    mov ds,ax
    mov ah,9h ;print message
    mov dx,offset msg1
    int 21h
    mov ax,4c00h ;exit program
    int 21h
cseg ends
end start
```

2.5.2 เปรียบเทียบระหว่างภาษาแอสเซมบลีกับภาษาเครื่อง {OOP}

Assembly & Machine Language

Assembly Language

```
ST 1, [801]
ST 0, [802]
TOP: BEQ [802], 10, BOT
      INCR [802]
      MUL [801], 2, [803]
      ST [803], [801]
      JMP TOP
BOT: LD A, [801]
      CALL PRINT
```

Machine Language

```
00100101 11010011
00100100 11010100
10001010 01001001 11110000
01000100 01010100
01001000 10100111 10100011
11100101 10101011 00000010
00101001
11010101
11010100 10101000
10010001 01000100
```

ทำงานใกล้ชิดกับฮาร์ดแวร์ เนื่องจาก Dependent หมายถึงเป็นภาษาที่ขึ้นอยู่กับเครื่อง

2.5.3 ภาษาชั้นสูง (High-level language) หรือภาษารุ่นที่ 3 (3GL:Third Generation Language)

- ถูกสร้างขึ้นมาเพื่อให้เขียนโปรแกรมได้ง่ายขึ้น โดยมีลักษณะเหมือนกับภาษาอังกฤษทั่วไป ผู้เขียนไม่จำเป็นต้องมีความรู้เกี่ยวกับฮาร์ดแวร์แต่อย่างใด ภาษานี้จำเป็นต้องมีตัวแปลภาษาเครื่องเช่นกัน เรียกตัวแปลนี้ว่า คอมไพเลอร์ (compiler) หรือ อินเตอร์พรีเตอร์ (Interpreter) อย่างใดอย่างหนึ่ง
- ตัวอย่างของภาษาชั้นสูง เช่น ภาษาปาสคาล ภาษาซี ภาษาโคบอล ภาษาเบสิก ภาษาฟอร์แทรน

2.5.3.1 ภาษา Pascal

{OOP}

ภาษาปาสกาล เป็นภาษาโปรแกรมที่ใช้กันอย่างกว้างขวาง โดยเฉพาะในวงการศึกษาคิดค้นขึ้นโดยนิกเลาส์ เวียร์ท (Niklaus Wirth) นักวิทยาการคอมพิวเตอร์ชาวสวิสใน ค.ศ. 1970 เพื่อช่วยในการเรียนการสอนการเขียนโปรแกรมโครงสร้าง (structured programming) ภาษาปาสกาลนั้นพัฒนาขึ้นมาจากภาษาอัลกอล (Algol) และชื่อปาสกาลนั้นตั้งเพื่อเป็นเกียรติแก่แบลซ ปัสกาล นอกเหนือจากภาษาปาสกาลแล้ว เวียร์ทได้พัฒนาภาษาโมดูลากู (Modula-2) และโอเบอร์อน (Oberon) ซึ่งมีโครงสร้างคล้ายกับภาษาปาสกาล แต่สามารถรองรับการเขียนโปรแกรมเชิงวัตถุ (object-oriented programming)



```
Turbo Pascal 7.0
File Edit Search Run Compile Debug Tools
\WORK.PAS
Program Input:
uses crt;
var ID : string;
    Name, Surname : string;
    Jitpisai, Midterm, Final : integer;
    Avg sun : real;
begin
  clrscr;
  write('Input ID = ');readln(ID);
  write('Input Name = ');readln(Name);
  write('Input Surname = ');readln(Surname);
  write('Input Jitpisai = ');readln(Jitpisai);
  write('Input Midterm = ');readln(Midterm);
  write('Input Final = ');readln(Final);
  write('Input Grade = ');readln(Grade);
  write('Input Avg sun = ');readln(Avg sun);
  readln;
end.
```

F1 Help F2 Save F3 Open Alt+F9 Compile F9 Make

```
Start here 8ds.c X
1  #include<conio.h>
2  #include<math.h>
3  /*file name ex_func5.c*/
4  void main ()
5  {
6  float r ,pi = 3.1415;
7  float x = 9.0,y = 3.0;
8  r = pi/180;
9  printf(" *Math Function*\n");
10 printf("\n");
11 printf("sin(45/180) = %.4f\n",sin(r));
12 printf("cos(45/180) = %.4f\n",cos(r));
13 printf("tan(45/180) = %.4f\n",tan(r));
14 printf("\n \n");
15 printf("sqrt(9) = %.2f\n",sqrt(x));
16 printf("pow(9,3) = %.2f\n",pow(x,y));
17 printf("ceil(3.14) = %.2f\n",ceil(pi));
18 }
19
```

แบบฝึกหัด
***ให้นักศึกษาแปลคำสั่งนี้ลงในกระดาษ
*** แปลทุกบรรทัด
*** 15 นาที

ภาษา C++

{00P}

นำเข้าไลบรารี conio.h และ math.h

นำเข้าไลบรารี math.h เพื่อเรียกใช้ฟังก์ชันหรือคำสั่งที่เกี่ยวข้องกับคณิตศาสตร์

ส่วนเริ่มโปรแกรมหลัก ที่ทำงานแล้วไม่คืนค่า (void)

ประกาศตัวแปร r เป็นชนิด float
Pi เป็นชนิด float และเก็บค่าเริ่ม 3.1415

```
Start here 8ds.c x
1  #include<conio.h>
2  #include<math.h>
3  /*file name ex_func5.c*/
4  void main ()
5  {
6  float r ,pi = 3.1415;
7  float x = 9.0,y = 3.0;
8  r = pi/180;
9  printf(" *Math Function*\n");
10 printf("\n");
11 printf("sin(45/180) = %.4f\n", sin(r));
12 printf("cos(45/180) = %.4f\n", cos(r));
13 printf("tan(45/180) = %.4f\n", tan(r));
14 printf("\n \n");
15 printf("sqrt(9) = %.2f\n", sqrt(x));
16 printf("pow(9,3) = %.2f\n", pow(x,y));
17 printf("ceil(3.14) = %.2f\n", ceil(pi));
18 }
19
```

```
ming\Content\15w\break.cpp - Dev-C++ 5.11
Project Execute Tools AStyle Window Help
IDM-GCC 4.9.2 32-bit Release
break.cpp
1 #include <iostream>
2 using namespace std;
3
4 int main ()
5 {
6     for (int n = 1; n <= 10; n++) {
7         if(n == 5) break;
8         cout << n << ", ";
9     }
10    cout << " end loop";
Compile Log Debug Find Results
```

```
#include<stdio.h>
#include<conio.h>
void main()
{
    int a,b;
    printf("enter a: ");
    scanf("%d",&a);
    printf("enter b:");
    scanf("%d",&b);
    printf("%d+%d=%d",a,b,a+b);|
    getch();
}
```

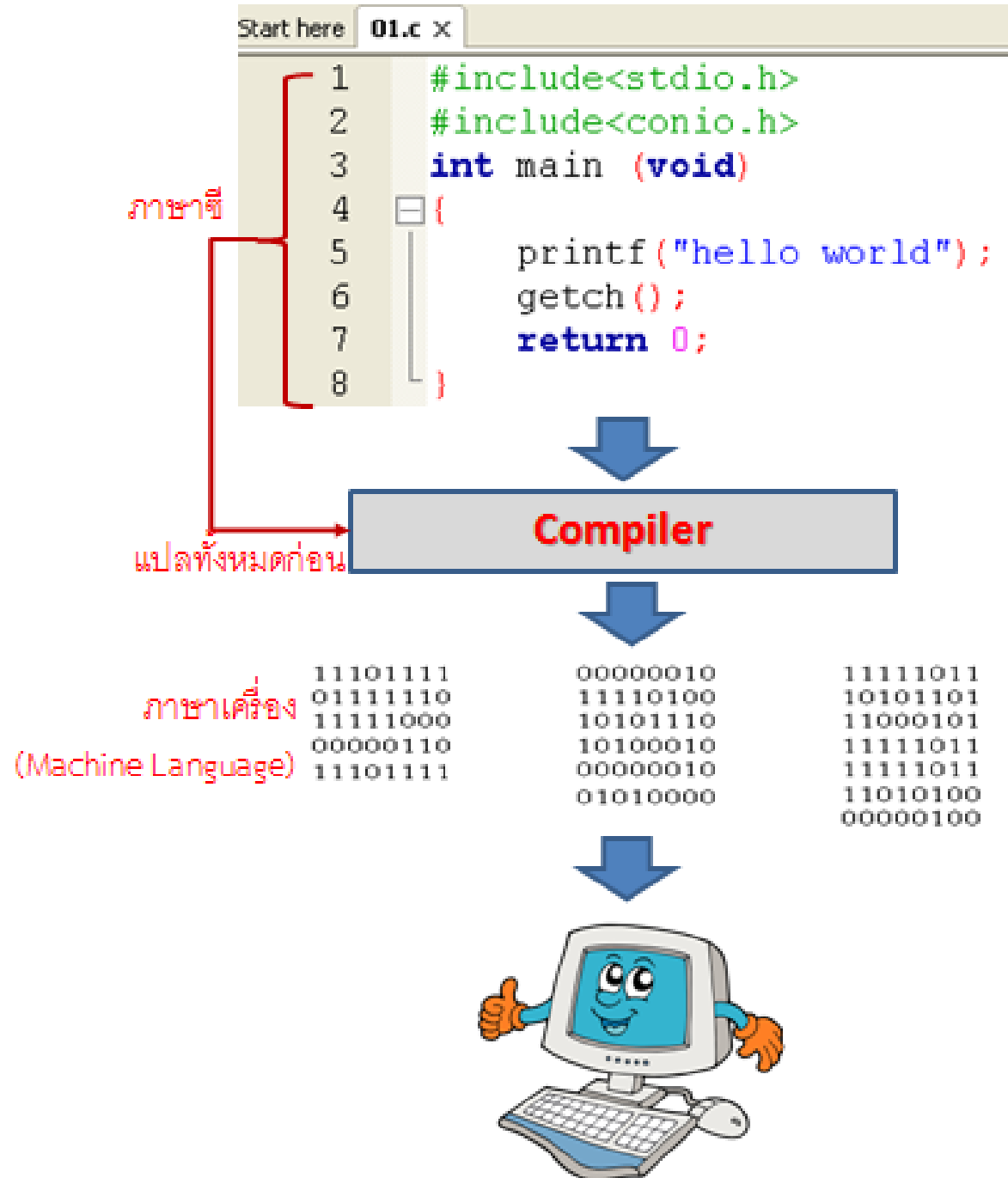
2.5.4 ภาษาชั้นสูงมาก (Very high-level language) หรือภาษารุ่นที่ 4 (4GL)

- เป็นภาษาที่มีลักษณะคล้ายภาษาพูดตามปกติของมนุษย์ ภาษานี้จะช่วยให้การเขียนโปรแกรมเร็วมากขึ้นกว่าภาษาในรุ่นที่ 3 เนื่องจากมีเครื่องมือที่ช่วยในการสร้างแบบฟอร์มหน้าจอ เพื่อจัดการกับข้อมูลรวมไปถึงการออกรายงาน เมนูต่าง ๆ
- ตัวอย่างของภาษาชั้นสูงมากได้แก่ informix-4GL, MAGIC , Delphi , Power Builder ฯลฯ

2.5.5 ภาษาธรรมชาติ (Natural language) หรือภาษารุ่นที่ 5 (5GL)

- เป็นภาษาที่สามารถสั่งงานคอมพิวเตอร์โดยใช้รูปแบบของภาษามนุษย์ได้เลย คำสั่งอยู่ในรูปแบบที่ไม่แน่นอนตายตัว แต่คอมพิวเตอร์จะทำการแปลให้ออกมาในรูปแบบที่คอมพิวเตอร์เข้าใจได้ ภาษานี้ถูกสร้างขึ้นมาจากเทคโนโลยีทางด้านระบบผู้เชี่ยวชาญ (Expert system)
- ตัวอย่างภาษาในรุ่นที่ 5 ได้แก่ ภาษา PROLOG เป็นต้น

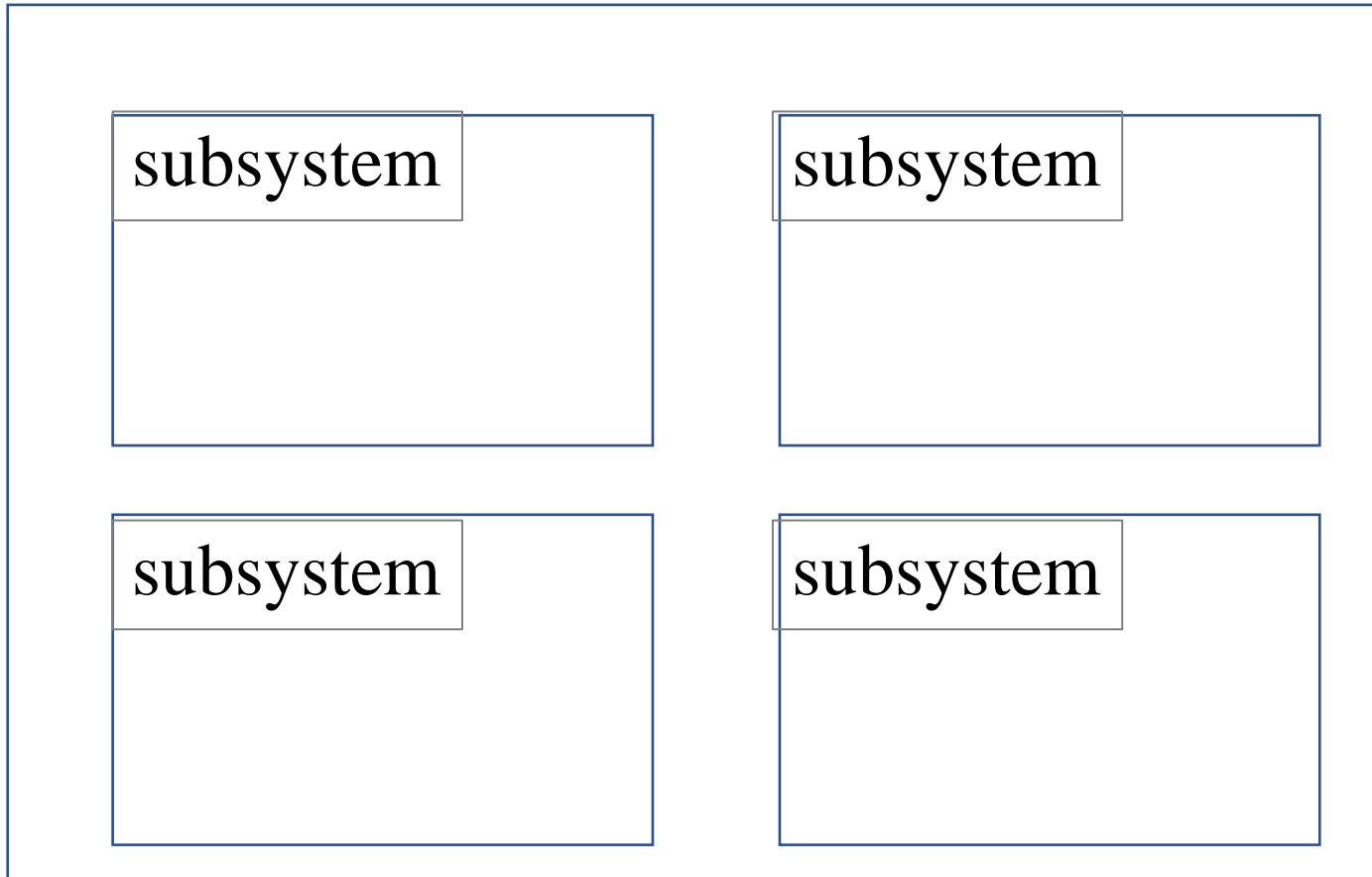
Compiler



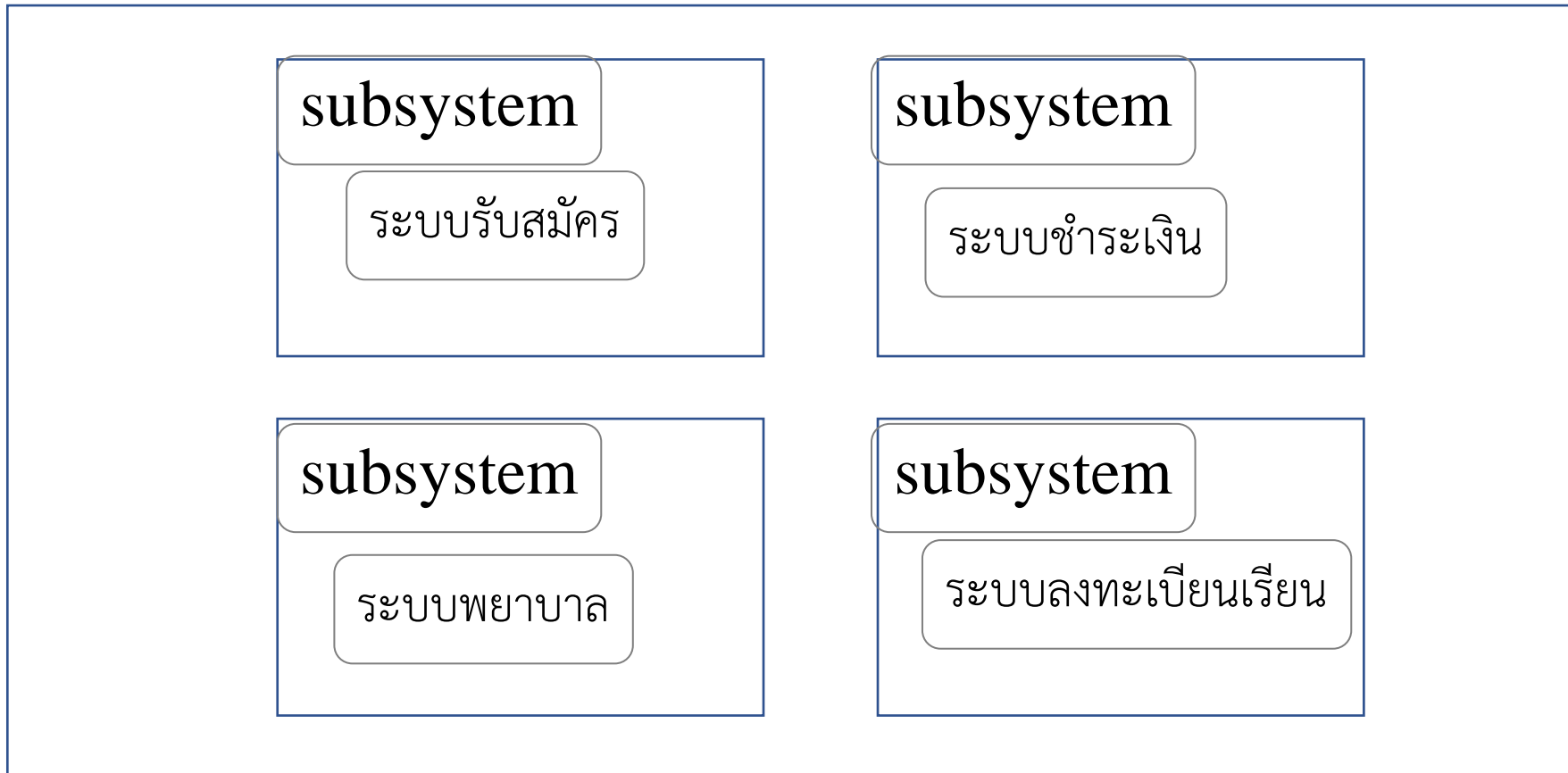
- DOS : Text Mode c, C++ , pascal , basic , qwbasic , cobol
- Windows : WinApp : Windows Application
 - Desktop Application : Visual Basic , Visual C# , Visual C++ , Dephi , Java ฯลฯ
- Web : Web App : PHP , JSP , ASP
- Mobile : Mobile Application : Andriod Java , iOS Objective C, swift
- Cross platform : iOnic , google flutter ฯลฯ
- Web Mobile : เป็นเว็บ responsive web

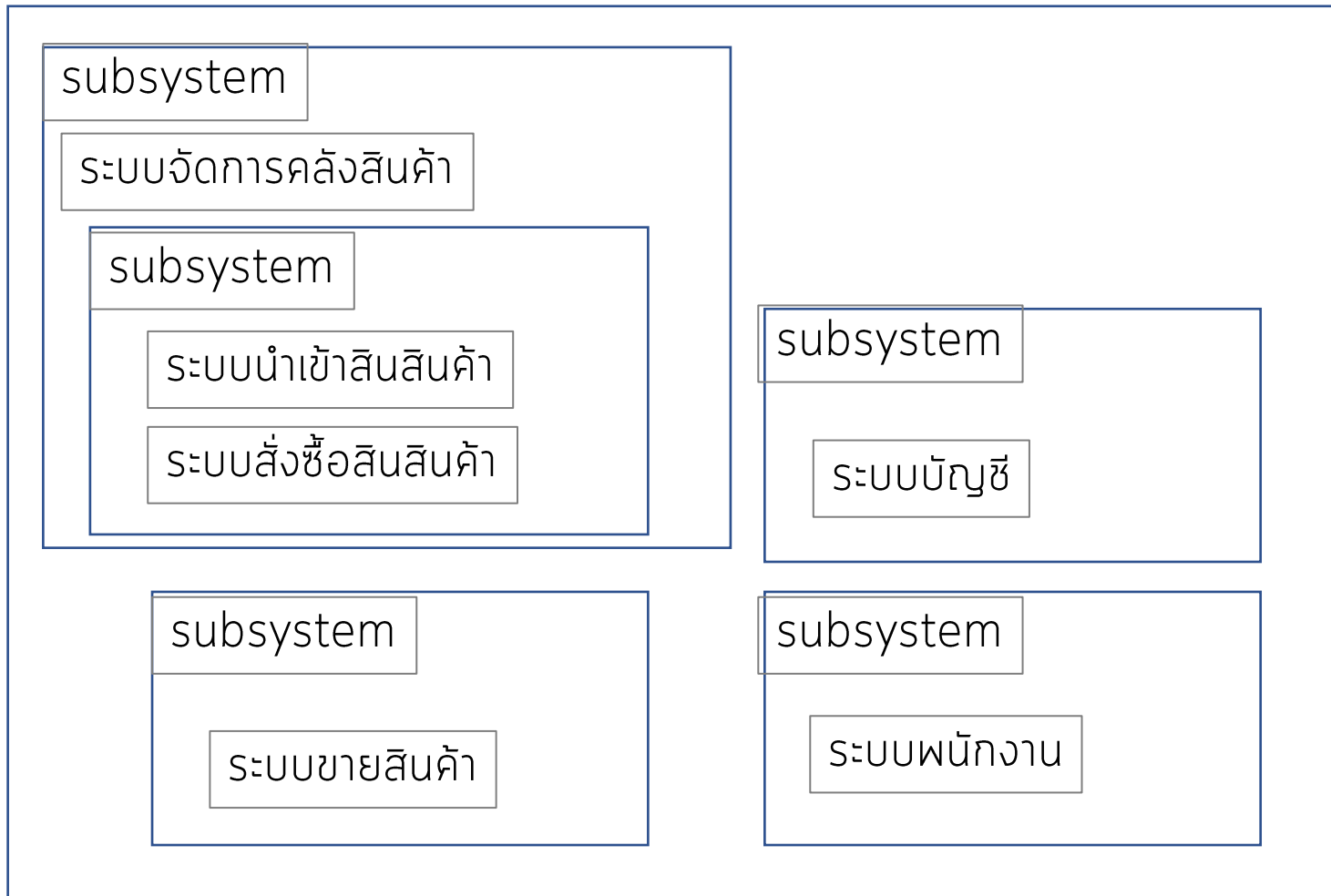
- TPS : Transaction Processing Unit ระบบประมวลผลรายการ เช่น ระบบฝากเงิน -ถอนเงิน ชื้อมา ขายไป ระบบลงทะเบียน
- MIS : Management Information System ระบบสารสนเทศเพื่อการจัดการ เช่น ระบบห้องสมุด ระบบคลังสินค้า ระบบการเรียนการสอน ระบบงานธนาคาร ระบบจัดซื้อจัดจ้าง ระบบบริหารงานบุคคลากร ระบบบริหารโรงแรม ระบบบริหารศูนย์บริการรถยนต์ ฯลฯ
- DSS : Decision Support System ระบบสนับสนุนการตัดสินใจ การแผนลงทุนขยายกิจการ การตัดสินใจสร้างผลิตภัณฑ์ใหม่ ฯลฯ
- EIS : Executive Information System ระบบสนับสนุนผู้บริหาร เน้นแผนภูมิ กราฟ ตาราง
- ES : Expert System ระบบผู้เชี่ยวชาญ นำ A.I. ใช้สำหรับคิด รับรู้แทนมนุษย์ เช่น ระบบช่วยวินิจฉัยโรค ระบบพยากรณ์อากาศ ฯลฯ

Integrated system



Integrated system : ระบบสารสนเทศเพื่อการจัดการการเรียนการสอน





ที่มา :

PM: Project Manager

Top-Down Design

Structure Chart

ระบบบริหารจัดการร้านขายกาแฟ ABC cafe'

Head programmer1

Head programmer2

Head programmer3

ระบบคลังสินค้า

ระบบขายสินค้า

ระบบพนักงาน

programmer1

programmer2

programmer3

programmer4

ระบบสั่งซื้อ
สินค้า

ระบบนำเข้า
สินค้า

ชำระเงิน

ระบบรับ
พนักงาน

ระบบจ่ายเงิน
เดือนพนักงาน

ออก
ใบเสร็จ

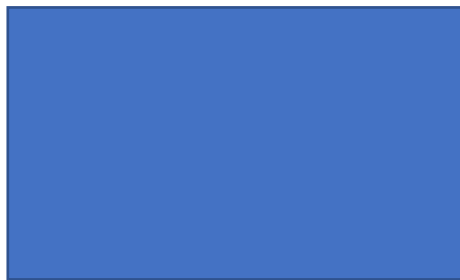
ตัวอย่าง การตั้งชื่อโครงการงานวิทยาการคอมพิวเตอร์ 2

- การพัฒนาระบบบริหารจัดการร้านชาวยกาแฟ ABC cafe'
- Development of

Web application

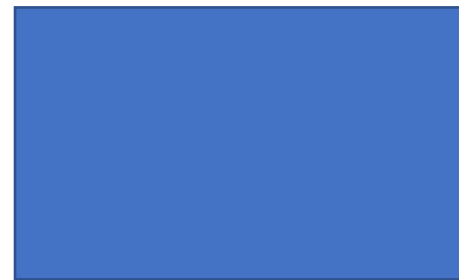
PHP, ASP , JSP , NodeJS, Python

JavaScript , Bootstrap , ฯลฯ



Back-end

Control + Management + Database
Admin



front-end

View, UXUI

ประเภทของข้อผิดพลาด (Error)

{00P}

- Syntax Error
- Runtime Error
- Logical Error

keyword Variables_name Data Type

VB

dim x as integer

VB

dim x as intDter

VB

dimxasinteger

ประธาน

+

กิริยา

+

กรรม

ประเภทของข้อผิดพลาด (Error)

Bug , Debug

- 0
- 1



Syntax Error

{00P}

- เป็นข้อผิดพลาดทางไวยากรณ์ของภาษา
- เกิดขึ้นง่ายและบ่อยที่สุด
- แก้ไขง่ายเนื่องคอมไพเลอร์ ของภาษาตรวจจับได้
- เกิด Human Error

```
Float x
```

```
int x =1.0;
```

```
float int;
```

```
short x = 5;
```

- ห้ามขึ้นต้นด้วยตัวเลข `int 2x;`
- ห้ามมีช่องว่าง เช่น `float x y;`
- ห้ามตั้งชื่อซ้ำกับคำสงวน reserved word เช่น `int int;` `int float;` `int for;` `int while;`
- ห้ามมีเครื่องหมาย หรือตัวดำเนินการ เช่น `int a+b;` `int x*y;`

- เป็นข้อผิดพลาดเมื่อโปรแกรมถูกทำงาน รันไทม์
- ผ่านการตรวจจับของคอมไพเลอร์
- ไม่มี syntax error

การหารด้วย 0

```
int x= 5, y = 0;
```

```
int z = x/y;
```

Logical Error

{00P}

- เป็นข้อผิดพลาดที่เกิดขึ้นจากการตีความหมายของโจทย์ผิด
- หรือ แปลความหมายของโจทย์ไม่ถูกต้อง ทำให้เขียนอัลกอรทึมผิดพลาด
- โปรแกรมทำงานได้ แต่ได้ผลลัพธ์ไม่ถูกต้อง

สรุป : Logical Error หมายถึงข้อผิดพลาดที่เกิดขึ้นในโปรแกรมเมื่อโค้ดทำงานแต่ไม่ได้ผลลัพธ์ตามที่คาดหวัง เป็นความผิดพลาดที่เกิดจากข้อผิดพลาดที่เขียนอยู่ในส่วนของตรรกะ (logic) ซึ่งอาจเกิดจากข้อผิดพลาดในวิธีการคำนวณหรือการประมวลผลข้อมูลในโปรแกรม

1. Algorithm วิธีการแก้ไขปัญหา
2. Syntax ไวยากรณ์ทางภาษา

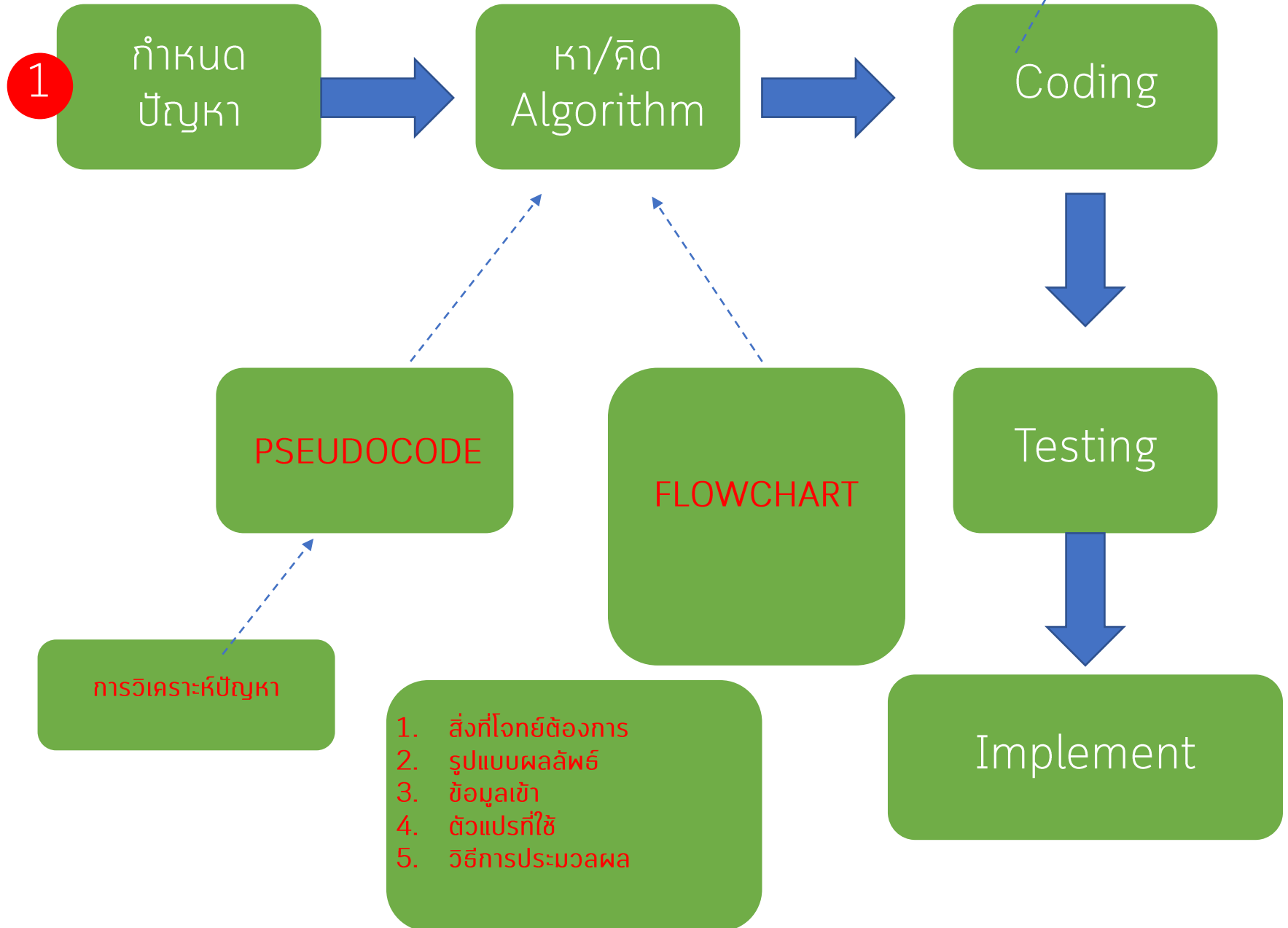
Algorithm
วิธีการแก้ไข
ปัญหา

Syntax
ไวยากรณ์ทาง
ภาษา

- เครื่องมือ(Tools) ในการนำเสนออัลกอริทึม ให้/ใช้สื่อสารกับบุคคลที่เกี่ยวข้อง
- หลังจากที่ได้อัลกอริทึมมาแล้วจะนำเสนอให้คนอื่น ๆ เข้าใจสามารถใช้เครื่องได้ดังนี้

- PSEUDO - CODE
- FLOWCHART

Syntax / Language / Tools



1. สิ่งที่ต้องพิจารณา

สิ่งที่ต้องพิจารณาคือ พ.ท. สี่เหลี่ยมผืนผ้า

2. รูปแบบผลลัพธ์

1. สิ่งที่ต้องพิจารณา
2. รูปแบบผลลัพธ์
3. ข้อมูลเข้า
4. ตัวแปรที่ใช้
5. วิธีการประมวลผล

dos



web



windows



Mobile



- INPUT
- PROCESS
- OUTPUT

DOS

กรุณาป้อน ความยาว :
กรุณาป้อน ความกว้าง :

DOS

พื้นที่ สีเหลี่ยมผืนผ้าเท่ากับ : xxx.xx

โปรแกรมคำนวณพื้นที่สี่เหลี่ยมผืนผ้า

ความยาว :

ผลลัพธ์

ความสูง :

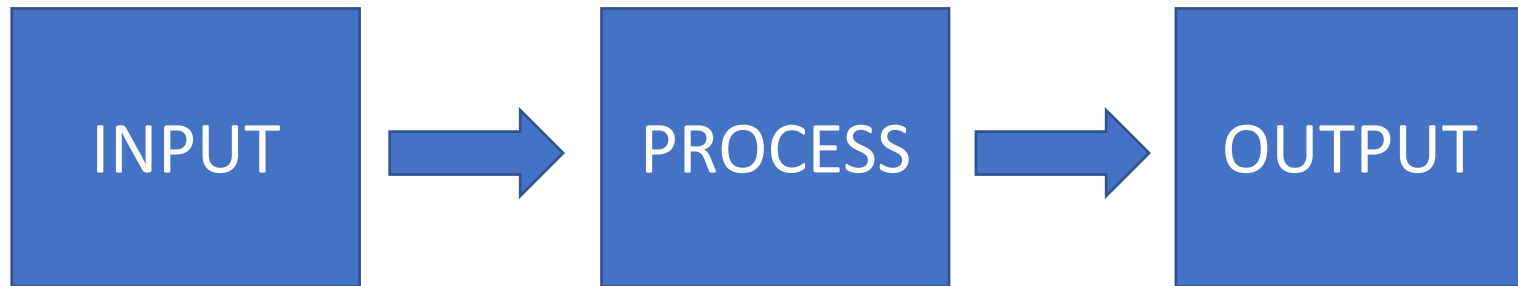
คำนวณ

ยกเลิก

3. ข้อมูลนำเข้า

{00P}

- ความกว้าง
- ความยาว



4. ตัวแปรที่ใช้

{00P}

- INPUT
 - Width แทน ความกว้าง
 - Height แทน ความสูง
- OUTPUT
 - RectangleArea แทน พื้นที่สี่เหลี่ยม
- PROCESS
 - RectangleArea แทน พื้นที่สี่เหลี่ยม

Width

Height



VB

Dim width , Height , RectangleArea

Dim x as int16
Dim x as String
Dim y as single
Dim width as single

4. ตัวแปรที่ใช้

{00P}

- INPUT
 - Width แทน ความกว้าง
 - Height แทน ความสูง
- OUTPUT
 - RectangleArea แทน พื้นที่สี่เหลี่ยม
- PROCESS
 - RectangleArea แทน พื้นที่สี่เหลี่ยม

Width

Height



Java

```
int width; , int Height;
```

Algorithm

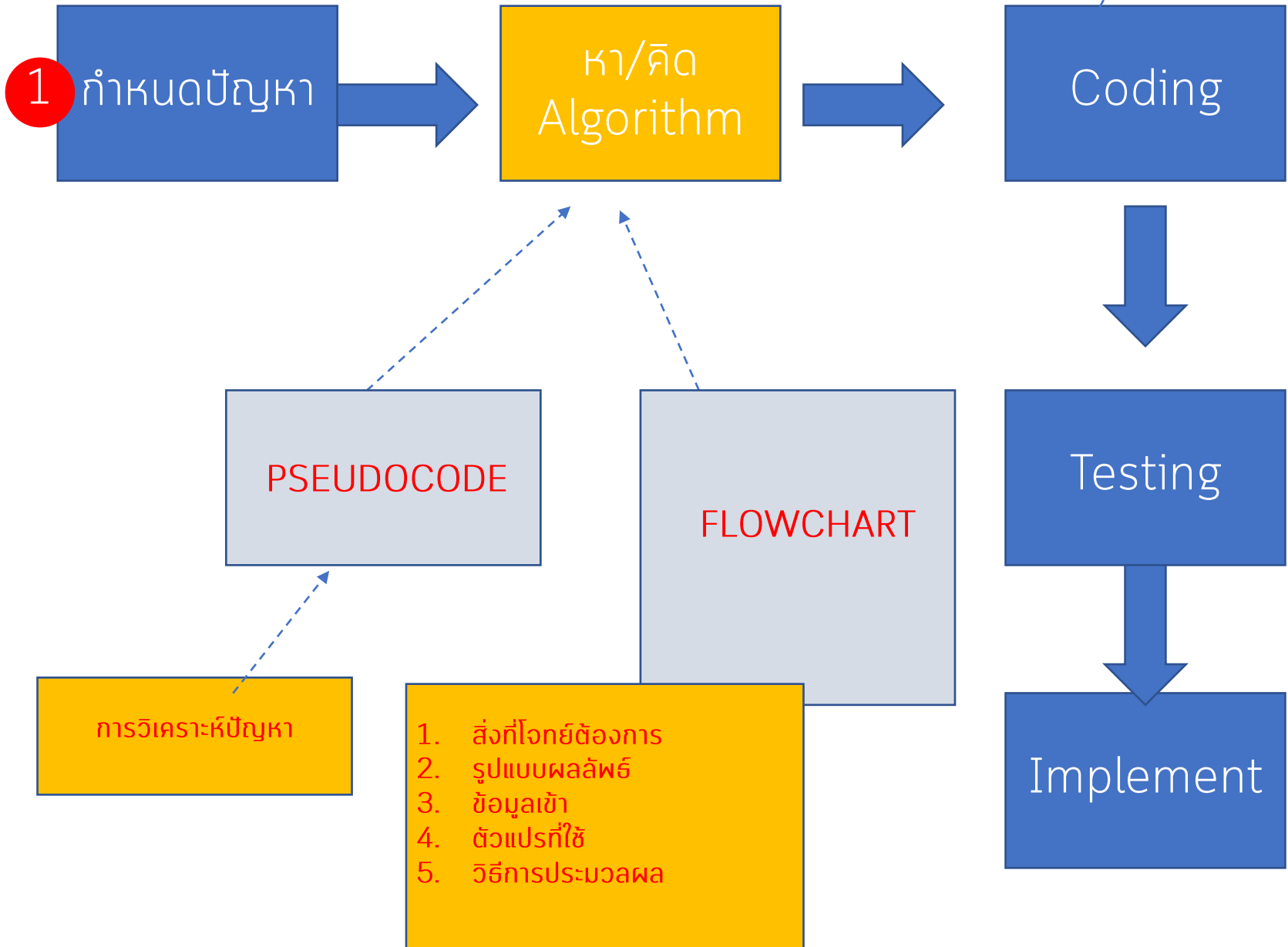
1. กว้าง*ยาว
2. ...
3. ...

1. กว้าง*ยาว
2. ...
3. ...

- จงวิเคราะห์ปัญหาของการหาค่าอายุเฉลี่ยของนักศึกษาทั้งห้องนี้
- ทำใส่ ms-word ส่งเข้าไลน์ กลุ่ม
- กำหนดส่งวันนี้ ก่อน 20.00 น.
- ใส่ชื่อ รหัส สาขา ตั้งเป็นชื่อไฟล์ : **แบบฝึกหัด_1_1_ชื่อ-นามสกุล_รหัส13
หลัก.docx**

ขั้นตอนการเขียนโปรแกรมคอมพิวเตอร์

1. การวิเคราะห์ปัญหา (Problem analysis)
2. การออกแบบโปรแกรม (Design)
3. การเขียนโปรแกรมโดยใช้ภาษาใดภาษาหนึ่ง (Coding)
4. การตรวจสอบข้อผิดพลาดของโปรแกรม (Testing and Debugging)
5. การทดสอบความถูกต้องของโปรแกรม (Testing and validating)
6. การทำเอกสารประกอบโปรแกรม (Documentation)
7. การบำรุงรักษาโปรแกรม (Program Maintenance)



ตัวอย่าง จงเขียนโปรแกรม เพื่อหา ว่า 1 ปี มีกี่วินาที

- วิเคราะห์ปัญหา ...

1. สิ่งที่ต้องต้องการ
2. รูปแบบผลลัพธ์
3. ข้อมูลเข้า
4. ตัวแปรที่ใช้
5. วิธีการประมวลผล

เขียนเป็นโปรแกรม ใน Visual Basic

เขียนเป็นโปรแกรม ใน Java

1. สิ่งที่โจทย์ต้องการ

- สิ่งที่โจทย์ต้องการ คือ **จำนวนวินาที ต่อ 1 ปี**

จงเขียนโปรแกรม เพื่อหา ว่า 1 ปี มีกี่วินาที

2. รูปแบบผลลัพธ์

DOS

จำนวนวินาทีใน 1 ปี เท่ากับ x,xxx

GUI

โปรแกรมคำนวณจำนวนวินาทีใน 1 ปี

จำนวนวินาทีใน 1 ปี เท่ากับ :

คำนวณ

ยกเลิก

3. ข้อมูลนำเข้า

- ไม่มีข้อมูลนำเข้า
- เนื่องจากโปรแกรมสามารถคำนวณค่าได้โดยไม่ต้องนำเข้าข้อมูล
- รู้ว่าใน 1 ปี มี 365 วัน
- รู้ว่า 1 วัน มีที่ ชั่วโมง 24
- รู้ว่า 1 ชั่วโมง 60 นาที
- รู้ว่า 1 นาที มี 60 วินาที

การออกแบบโปรแกรมที่ดี ต้องเป็นมิตรกับผู้ใช้ นั้น user friendly

4. ตัวแปรที่ใช้

SecondInYears แทน จำนวนวินาทีใน 1 ปี

5. วิธีการประมวลผล

1. เริ่มต้น
2. กำหนด `SecondInYear` $\leftarrow 0$
3. คำนวณ `SecondInYear` $\leftarrow 365*24*60*60$
4. แสดงผลลัพธ์
5. สิ้นสุด

\leftarrow เครื่องหมาย `assign value`
เป็นการนำค่าด้านขวามือ ใส่ในตัวแปรด้านซ้ายมือ

รหัสเทียม (PSEUDO-CODE)

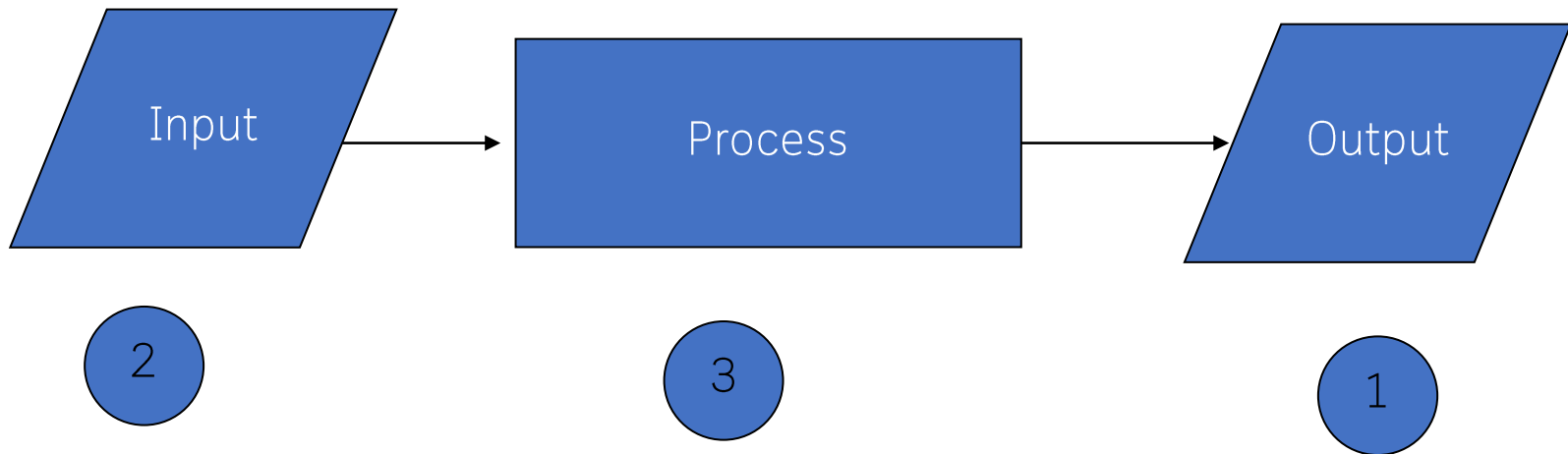
1. START / BEGIN
2. SET SecondInYear \leftarrow 0
3. COMPUTE SecondInYear \leftarrow 365*24*60*60
4. DISPLAY / PRINT / OUT SecondInYear
5. FINISH / STOP / END

จำนวนวินาทีใน 1 ปี เท่าไหร่

- INPUT
 - $365 * 24 * 60 * 60$
- PROCESS ?
 - Algorithm
- OUTPUT
 - คำตอบ

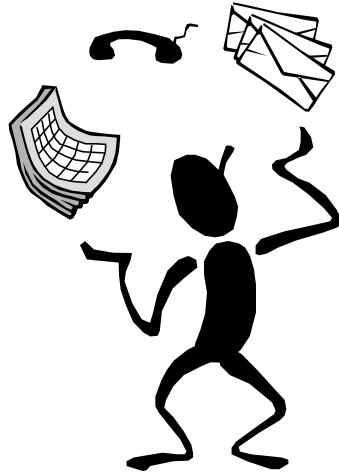
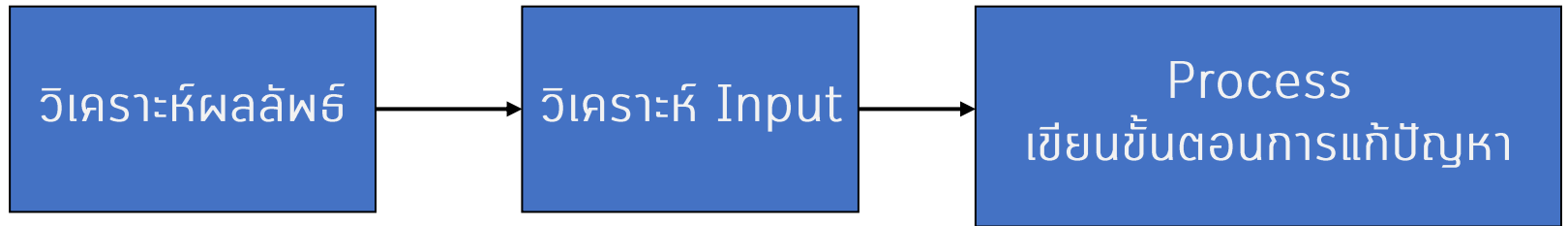
การวิเคราะห์ปัญหา

- สิ่งที่ต้องพิจารณา
 - Input ? (พิจารณาจาก Output)
 - Output? (พิจารณาอันดับแรก)
 - Process (ยังไม่ต้องสนใจตอนนี้)



แนวความคิดเบื้องต้นการวิเคราะห์ปัญหา

- การวิเคราะห์ปัญหา



การออกแบบโปรแกรม

□ เครื่องมือที่ใช้ในการออกแบบ

- ผังงาน (Flowchart) ขั้นตอนการแก้ปัญหาที่ละขั้นตอนในลักษณะรูปภาพ
- รหัสจำลอง (Pseudo) รูปแบบเป็นภาษาพูดง่าย ๆ ภาษาอังกฤษ หรือภาษาไทยก็ได้

รหัสเทียม (Pseudo-Code)

1. START / BEGIN /
2. SET SUM \leftarrow 0 , I \leftarrow 0
3. **WRITE PRINT / DISPLAY / SHOW**
4. INPUT / ENTER / READ
5. COMPUTE /SET I \leftarrow I + 1

6. STOP / END / FINISH

ASSIGN \leftarrow
Operator =

:=

การพัฒนาอัลกอริธึม

การพัฒนาอัลกอริธึม

- ข้อดีของรหัสเทียม (จำลอง)
 - แปลงเป็นโปรแกรมได้ง่าย
- ข้อดีของผังงาน
 - อ่านง่าย เข้าใจตรงกันได้

สัญลักษณ์ในผังงาน



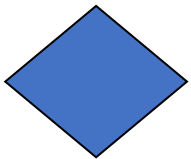
จุดเริ่มต้น หรือสิ้นสุด



อ่านข้อมูลเข้า หรือแสดงผล



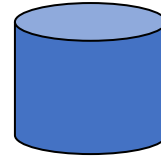
ประมวลผล



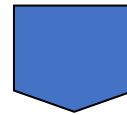
ตัดสินใจ



แสดงผลทางหน้าจอ



เพิ่มข้อมูล



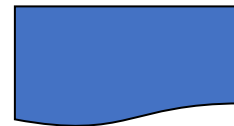
จุดเชื่อมต่อในหน้าอื่น



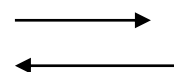
จุดเชื่อมต่อในหน้าเดียวกัน



โปรแกรมย่อย

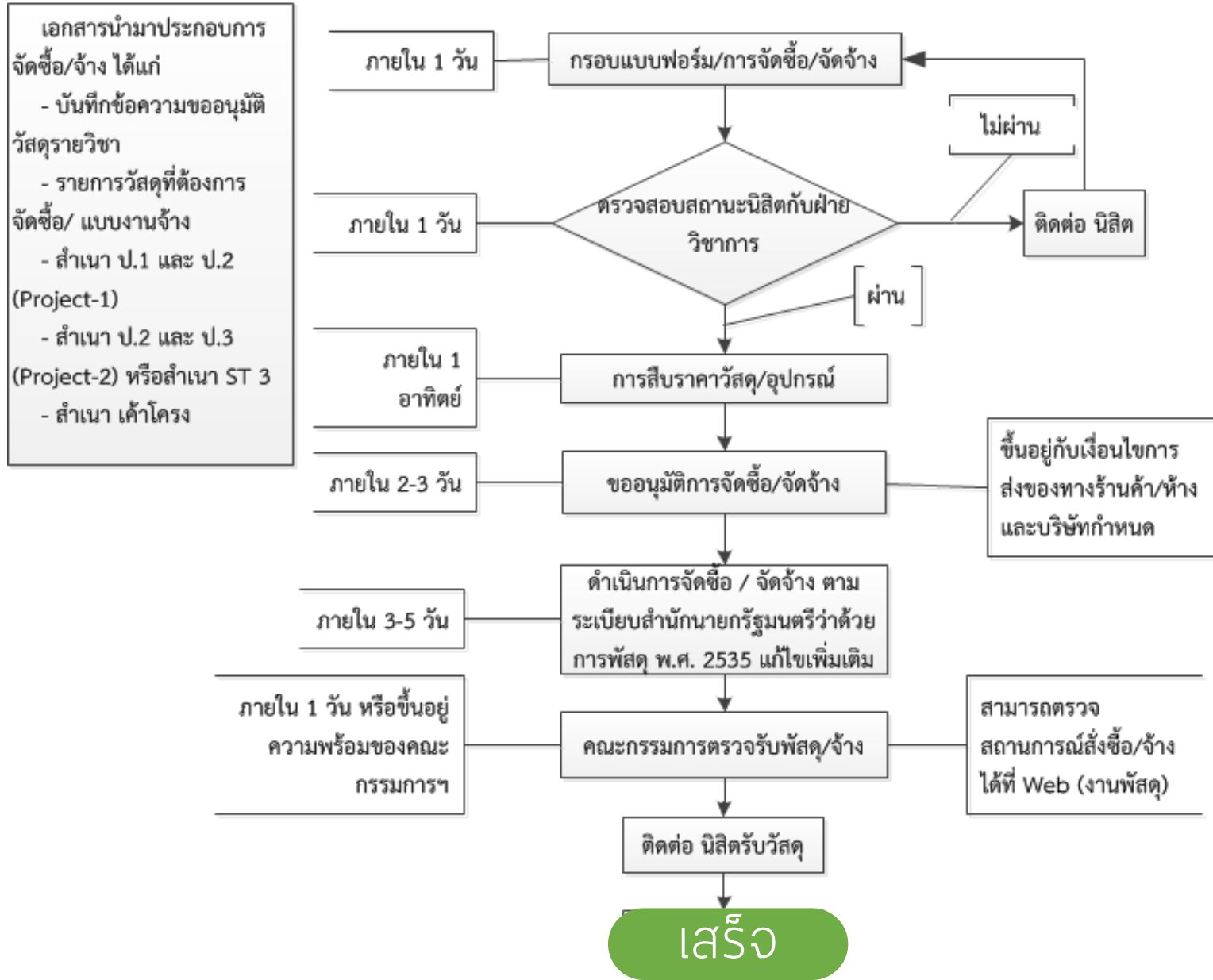


พิมพ์ผลทางเครื่องพิมพ์

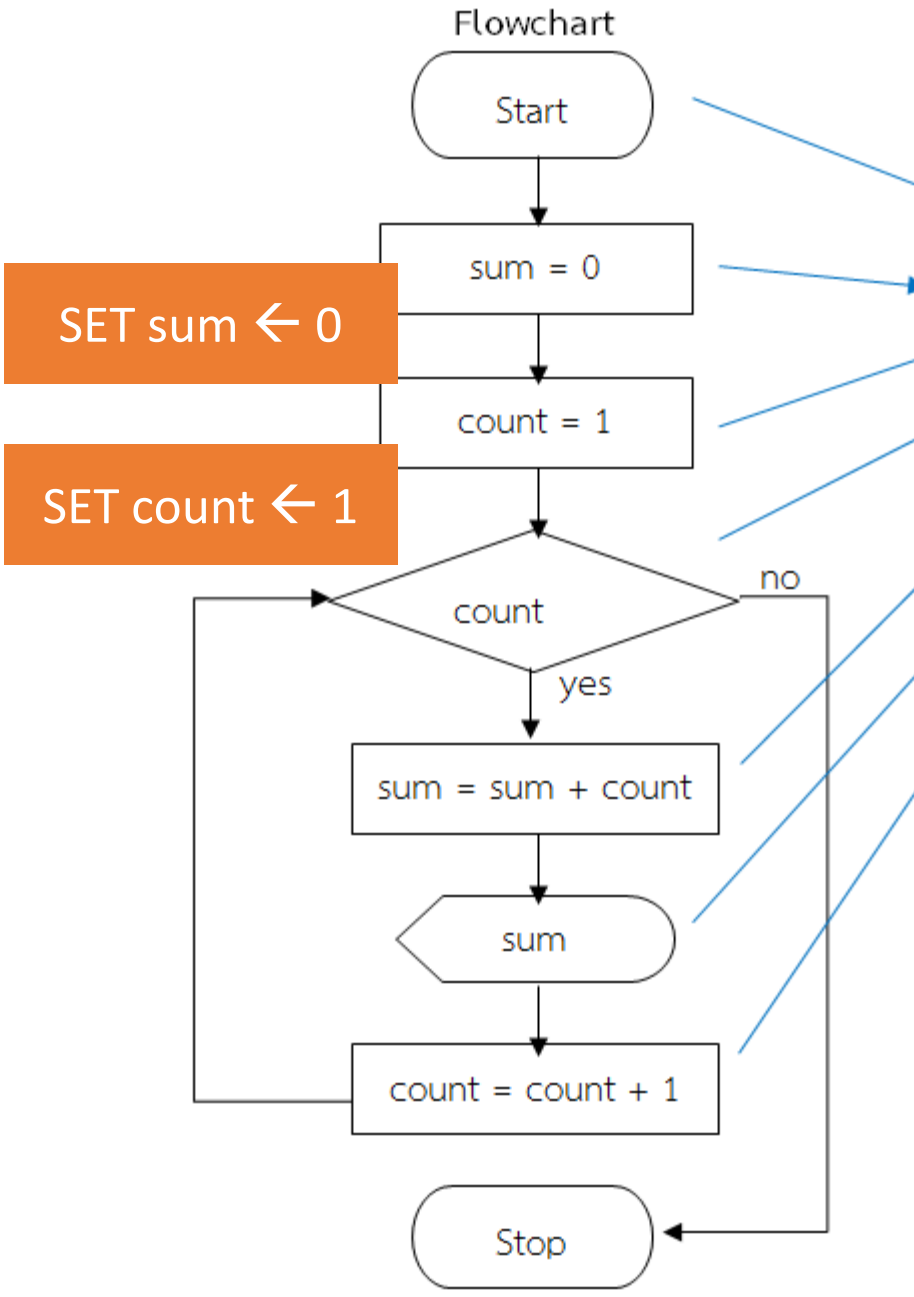


แสดงทิศทางการประมวลผล

ขั้นตอนการจัดซื้อ / จัดจ้าง รายวิชาปริญญาโท



1. จงวาดผังงานข้างต้นด้วย MS_word
2. จงแปลงเป็น PSEUDO-CODE เขียนลงในสมุดคู่มือ



SET sum ← 0

SET count ← 1

```

when clicked
  set sum to 0
  set count to 1
  repeat 10
    set sum to sum + count
    say sum for 1 secs
    change count by 1
  
```

****หมายเหตุ**
 - ใส่ say sum for 1 secs เพื่อให้เห็นลำดับการทำงานในแต่ละรอบ

PSEUDO-CODE
SET $X \leftarrow 4$

Pascal
 $X := 4;$

Vb
 $X = 4$

Java , c
 $X = 4;$

การพัฒนาอัลกอริธึม

- หลักการโครงสร้างควบคุม (Control structures)

การเขียนขั้นตอนในรายละเอียดของแต่ละอัลกอริธึม อาศัยโครงสร้างควบคุมดังนี้

- โครงสร้างลำดับ (Sequence structure)
- โครงสร้างตัดสินใจ (Selection structure)
- โครงสร้างทำซ้ำ (Repetition structure)

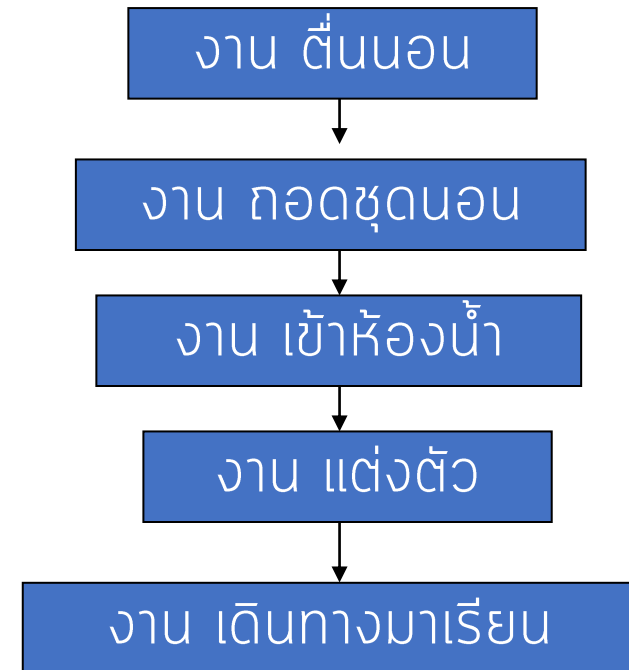
การพัฒนาอัลกอริธึม

• โครงสร้างลำดับ (Sequential)

- หมายถึง งานที่ต้องทำตามลำดับก่อนหลัง
- เช่น อัลกอริธึม “การตื่นนอนตอนรับวันใหม่”
 - งาน ตื่นนอน
 - งาน ถอดชุดนอน
 - งาน เข้าห้องน้ำ
 - งาน แต่งตัว
 - งาน เดินทางมาเรียน

ต้องจัดลำดับให้ถูกต้อง

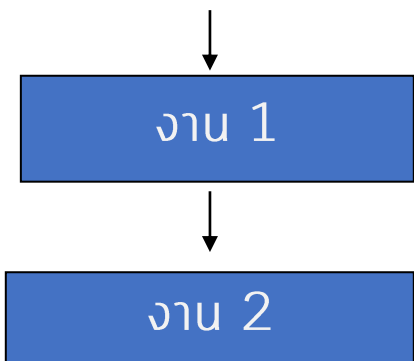
งาน เดินทางมาเรียน



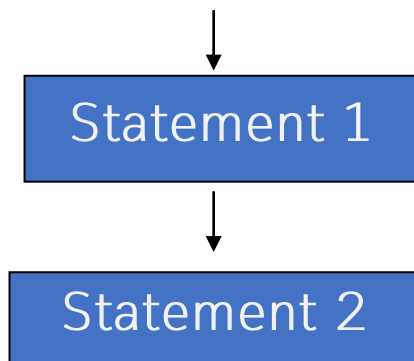
โครงสร้างการทำงานแบบลำดับ (Sequence)

- โครงสร้างลำดับ

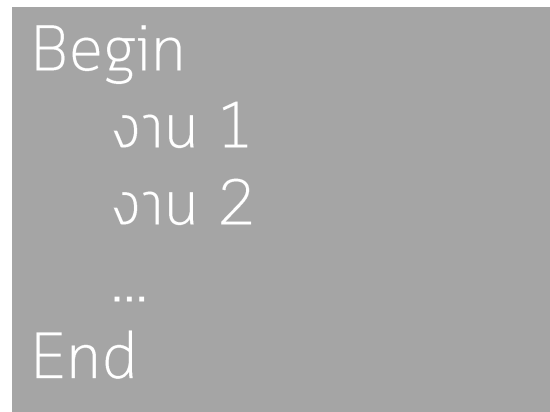
- หลักการให้มองงานเป็นชิ้นใหญ่ก่อน ยังไม่ต้องลงรายละเอียด แล้วจัดลำดับความคิดว่าจะทำอะไร ก่อนหรือหลัง ตามลำดับ



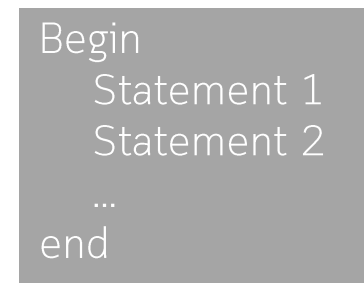
ผังงาน



flowchart



รหัสเทียม



ตัวอย่างการเขียนผังงาน

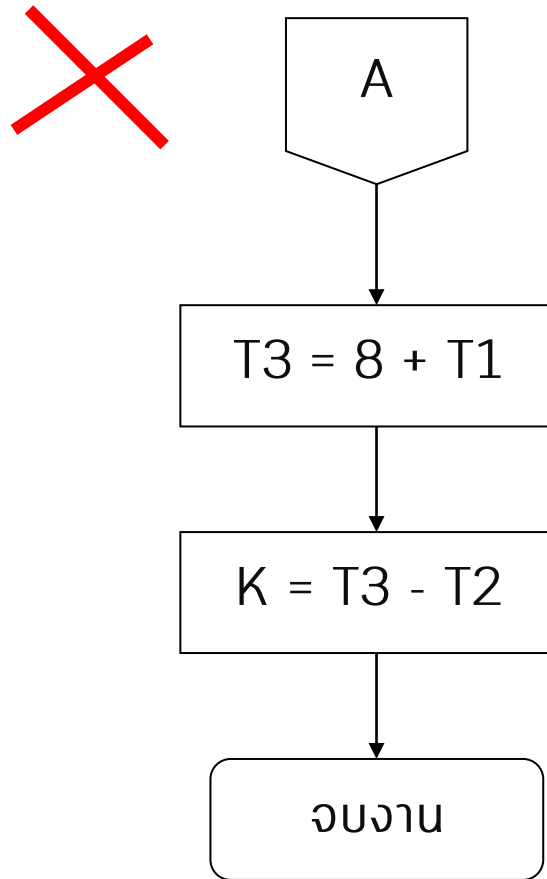
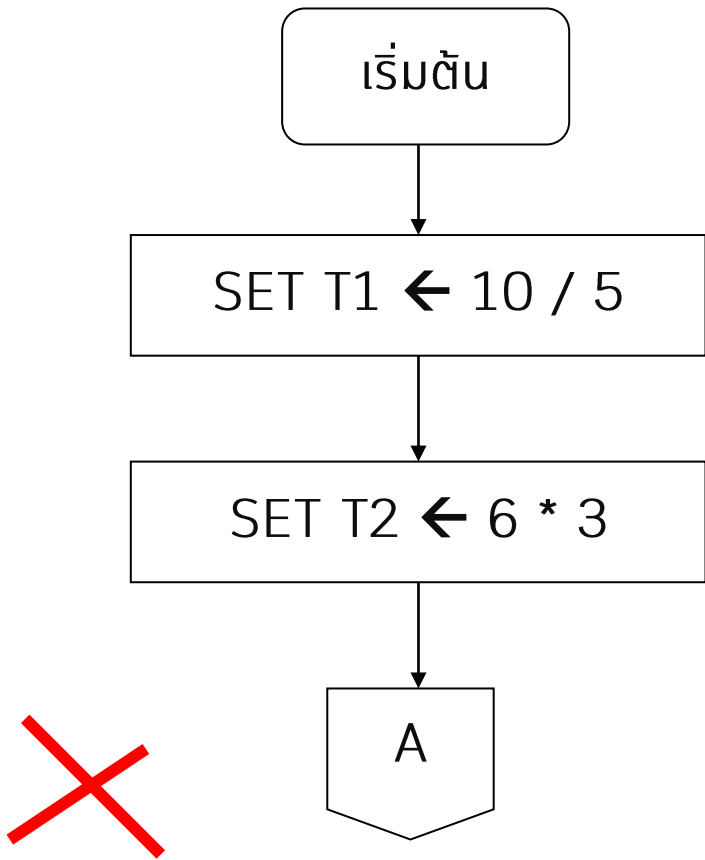
- จงเขียนผังงานที่แสดงลำดับการคำนวณตัวเลข เพื่อหาผลลัพธ์การคำนวณ แล้วเก็บผลลัพธ์ไว้ในหน่วยความจำตำแหน่ง K

$$8 + 10 / 5 - 6 * 3$$

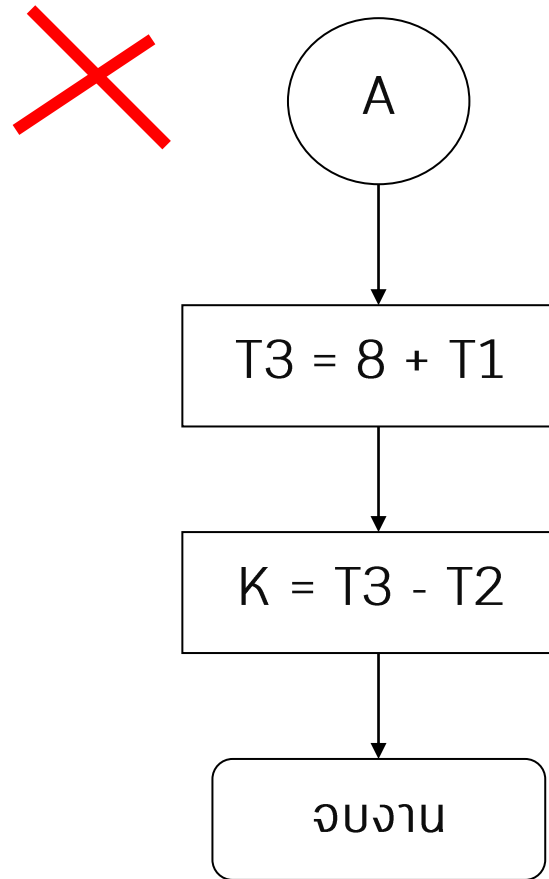
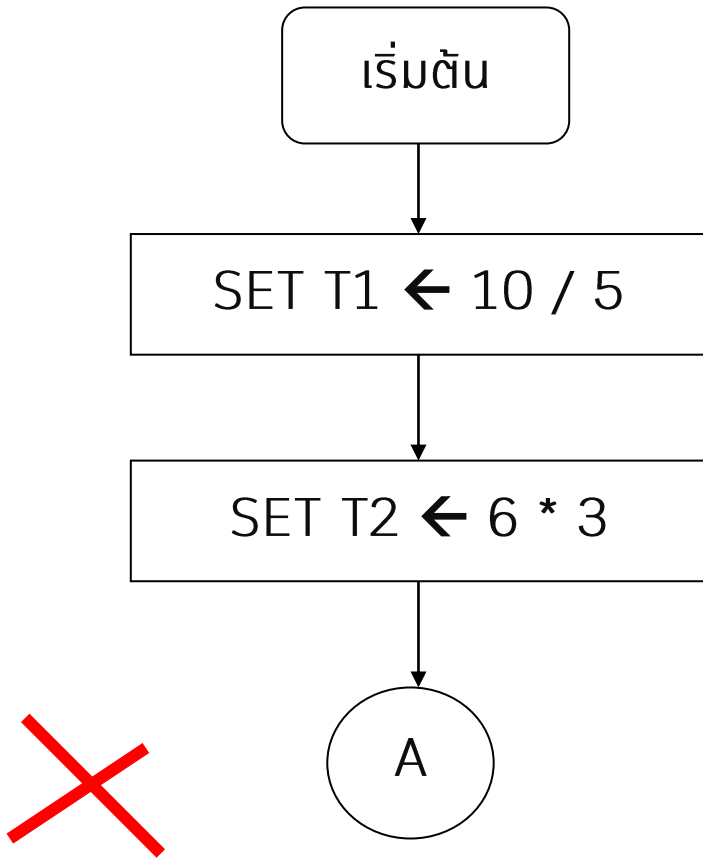
โดยมีลำดับการคำนวณทางเลขคณิต ดังนี้ (Precedence)

1. ยกกำลัง
2. คูณหรือหาร (โดยทำจากซ้ายไปขวา)
3. บวกหรือลบ (โดยทำจากซ้ายไปขวา)

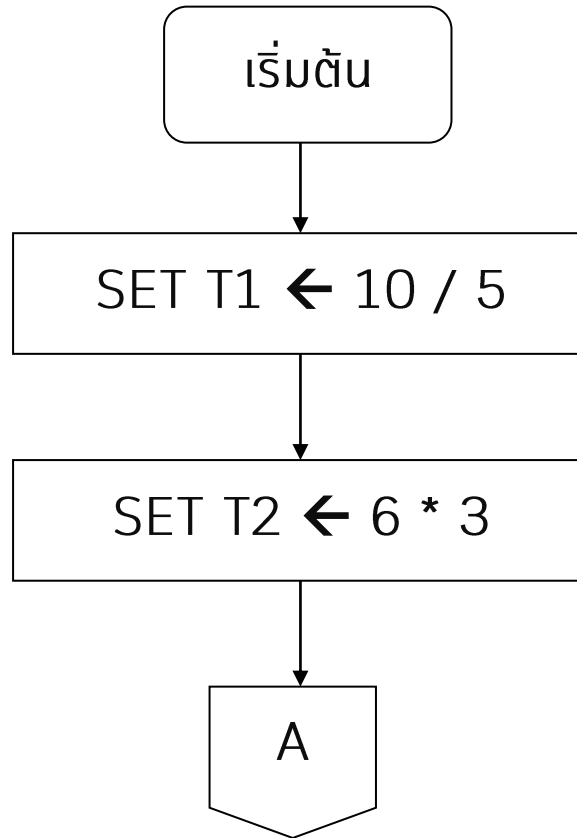
ตัวอย่างการเขียนผังงาน



ตัวอย่างการเขียนผังงาน

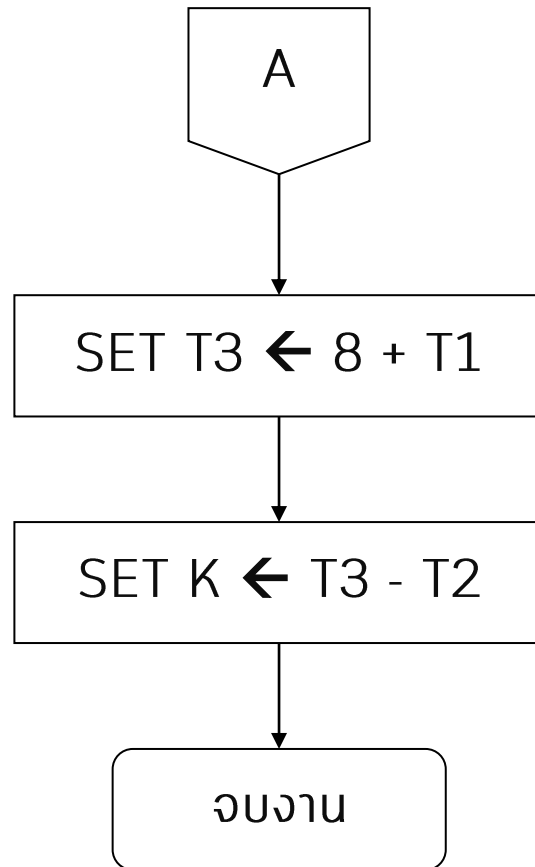


ตัวอย่างการเขียนผังงาน

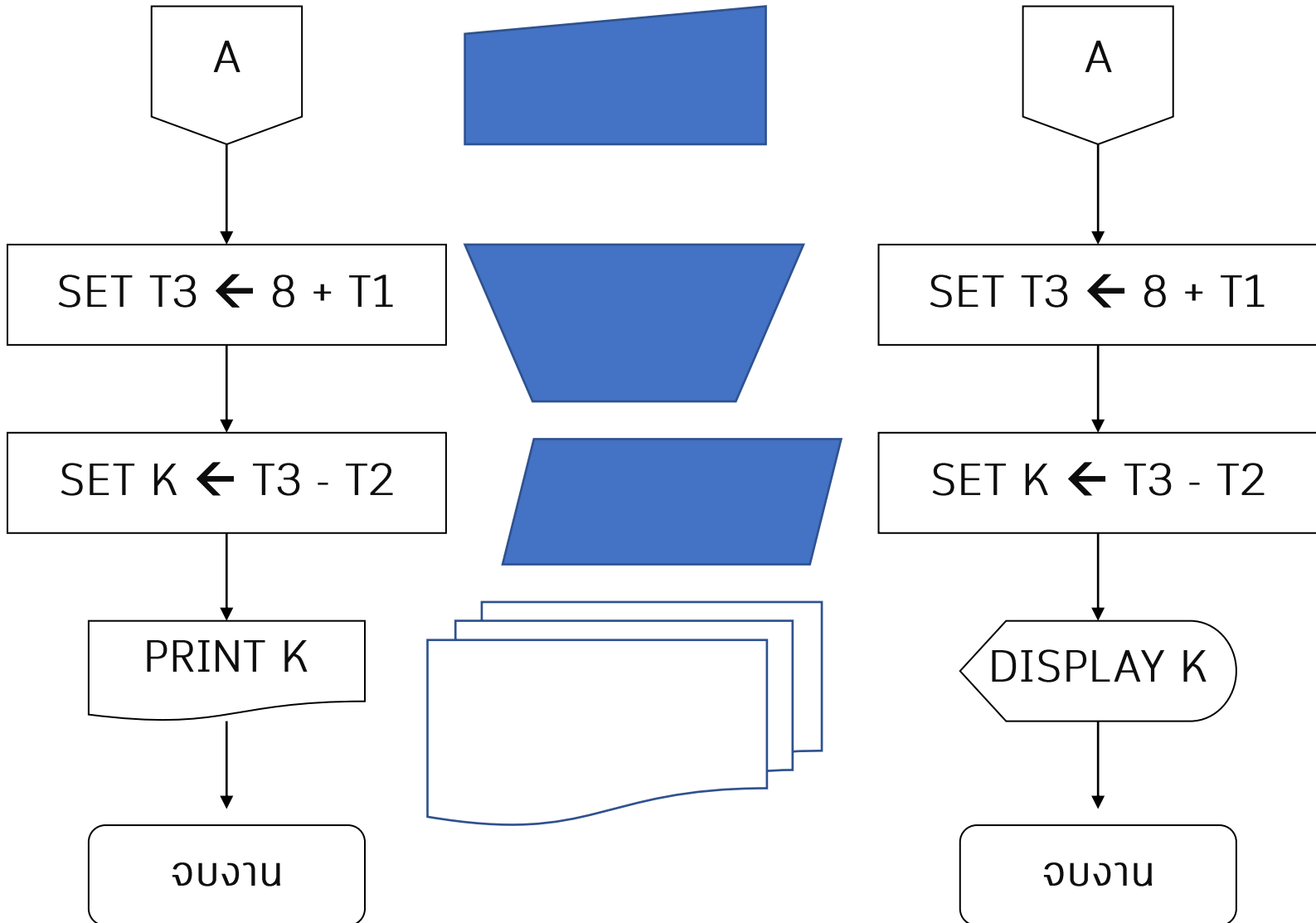


การเชื่อมต่อผังงานคนละหน้ากระดาษกัน ให้ใช้ รูปลิ้ม (connect)

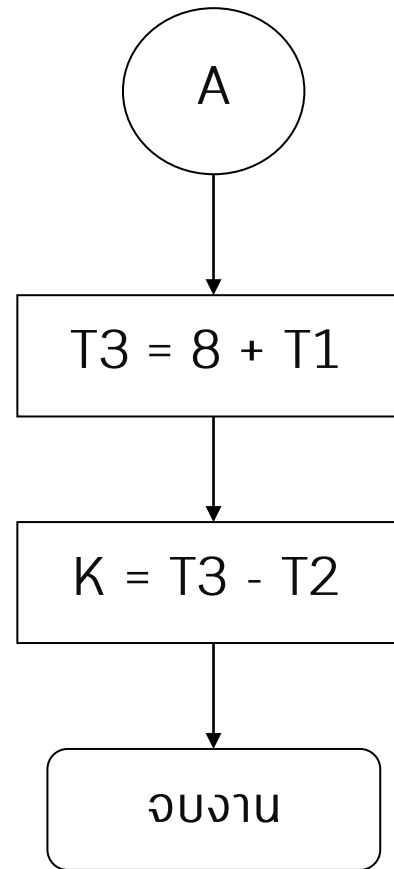
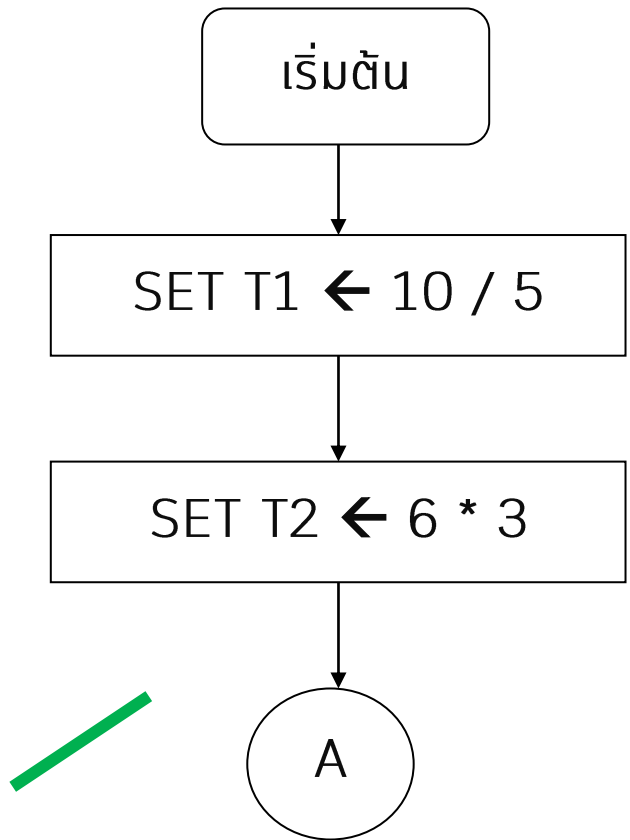
ตัวอย่างการเขียนผังงาน



ตัวอย่างการเขียนผังงาน



ตัวอย่างการเขียนผังงาน

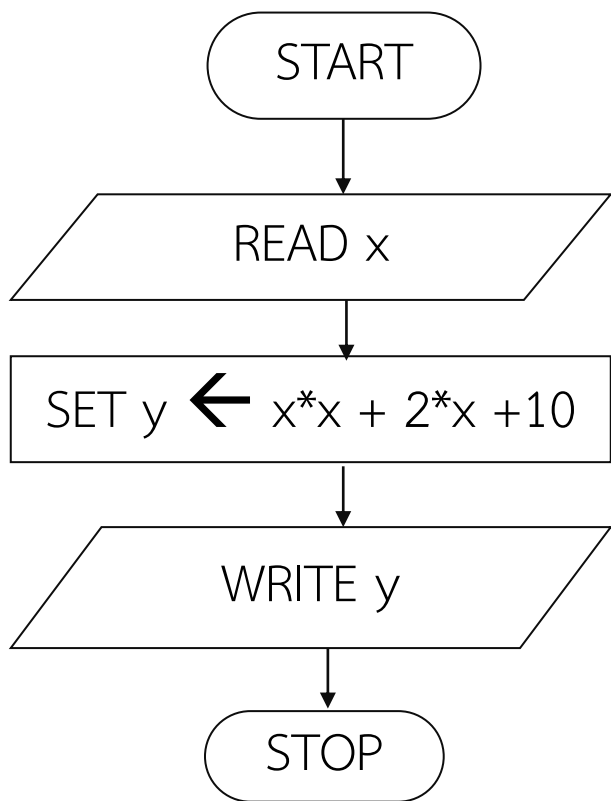


การเชื่อมต่อผังงานภายในกระดาษแผ่นเดียวกัน ให้ใช้ วงกลม (connect)

การพัฒนาอัลกอริธึม

example: จงเขียนโปรแกรมคำนวณค่า y ของสมการ

$$y = x^2 + 2x + 10$$



ผังงาน

BEGIN

READ x

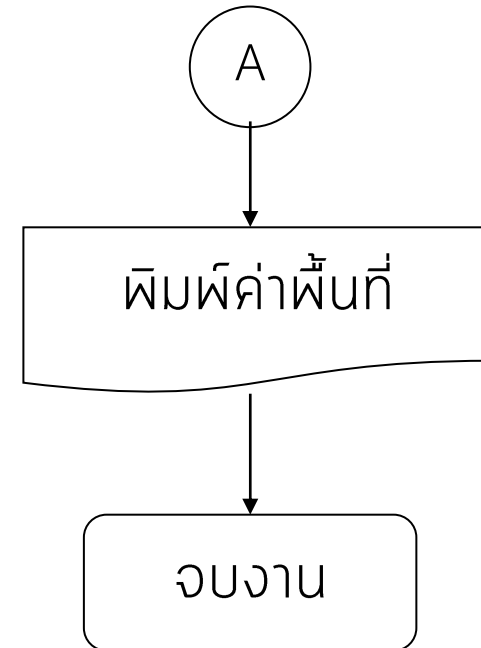
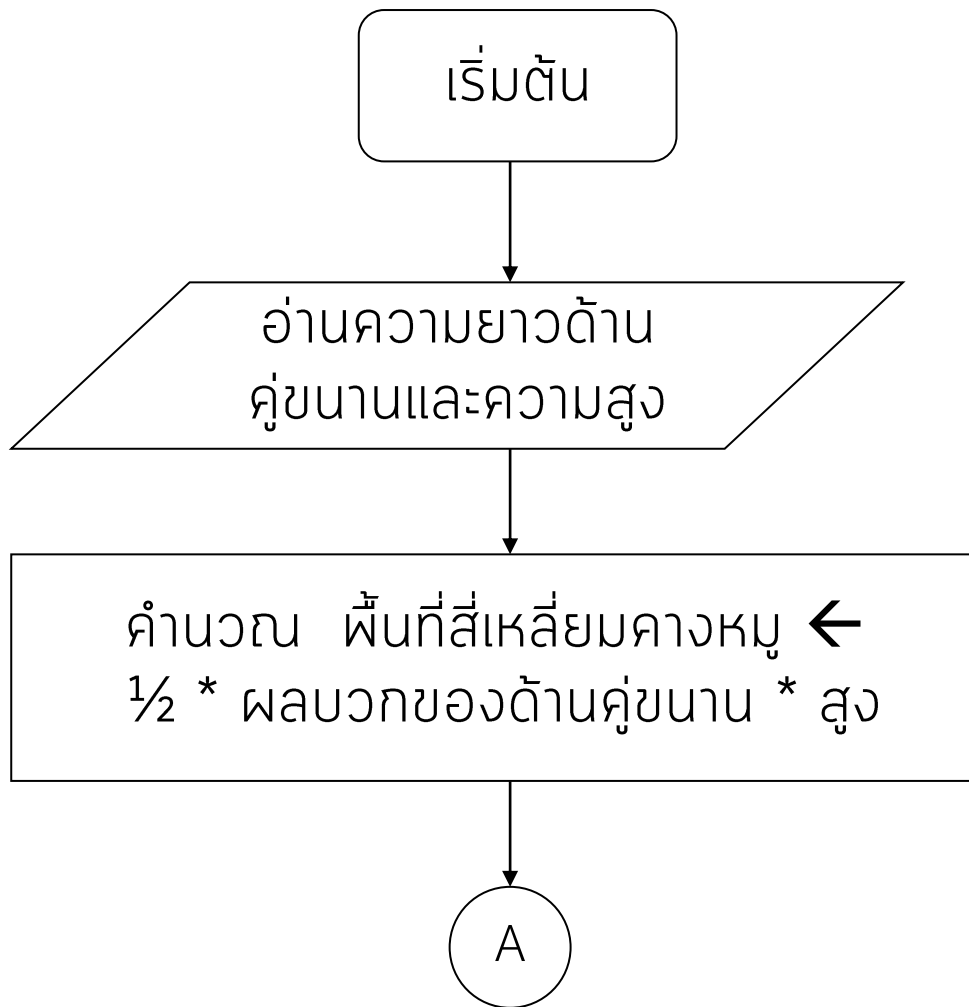
COMPUTE $y \leftarrow x*x + 2*x + 10$

DISPLAY / PRINT / OUT y

END

รหัสเทียม

จงเขียนผังงานคำนวณการหาพื้นที่สี่เหลี่ยมคางหมู โดยกำหนดค่าความยาวด้านคู่ขนานและความสูงให้



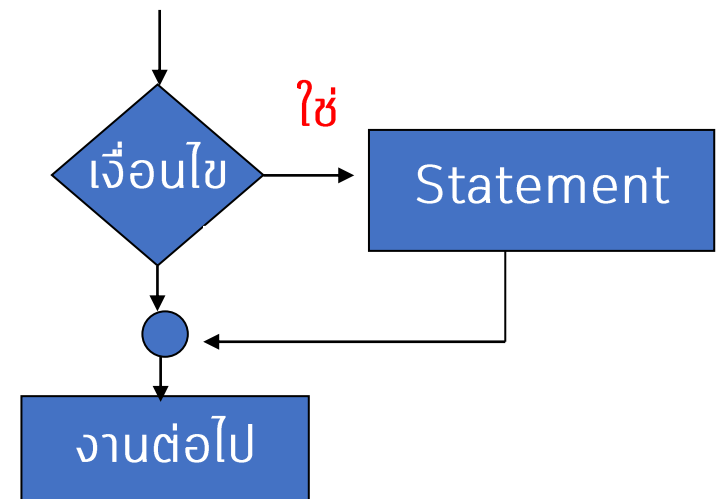
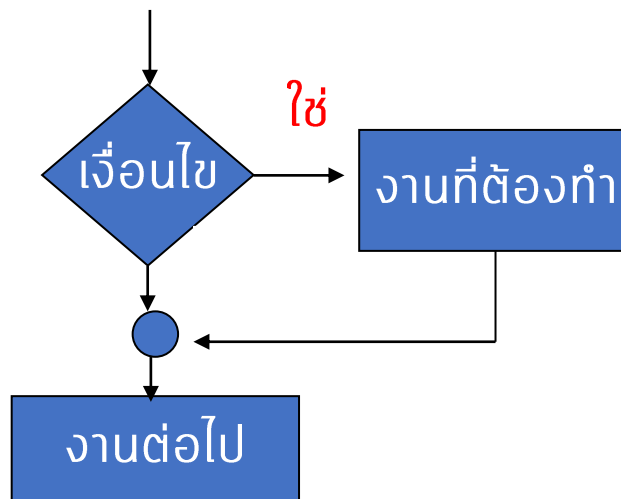
การพัฒนาอัลกอริธึม

- โครงสร้างตัดสินใจ (selection)
 - โครงสร้าง IF
 - โครงสร้าง IF else

การพัฒนาอัลกอริธึม

- โครงสร้าง IF

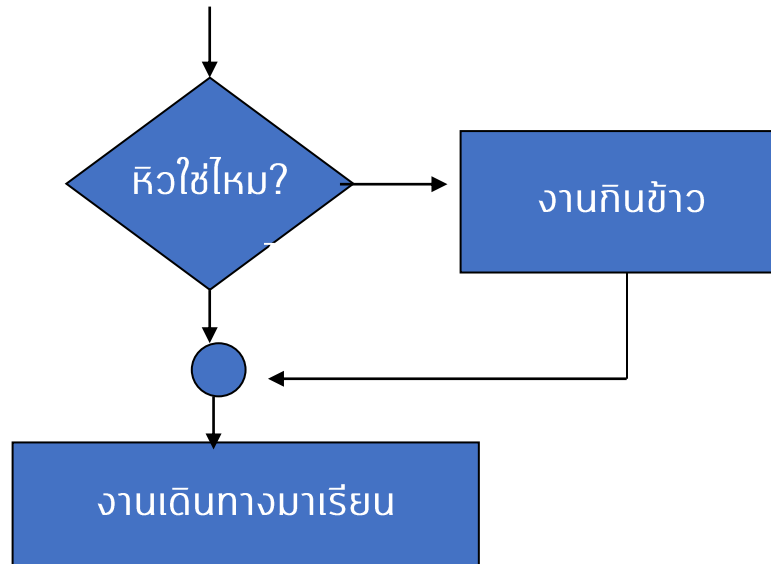
- เป็นโครงสร้างที่ทดสอบเงื่อนไข แล้วเลือกที่จะทำหรือไม่ทำ ก่อนที่จะไปทำงานอื่นต่อไป



การพัฒนาอัลกอริธึม

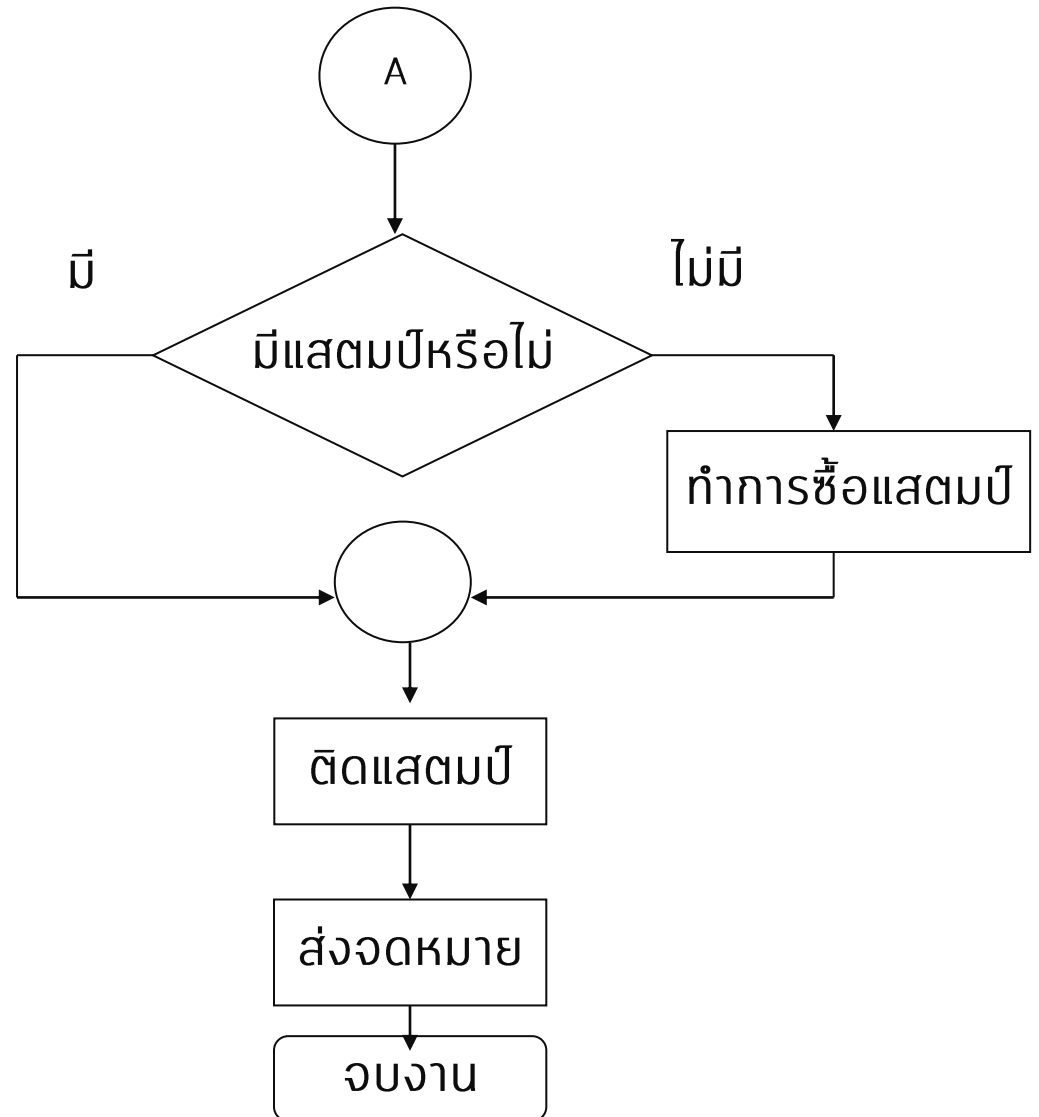
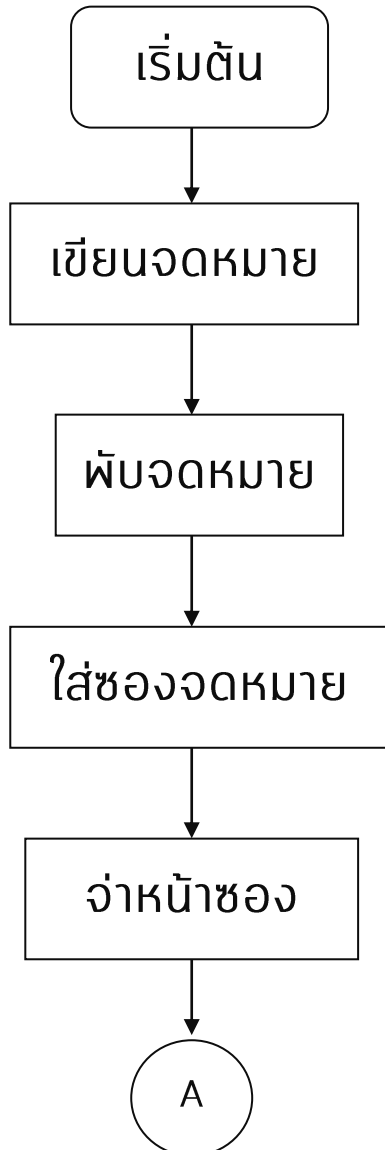
□ ตัวอย่าง โครงสร้าง IF

- ▣ แม่ถามว่าหิวข้าวหรือไม่ ถ้าหิวก็ให้กิน ถ้าไม่หิวไม่ต้องกิน แล้วจึงเดินทางไปโรงเรียน (งานต่อไป)



ตัวอย่างการเขียนผังงาน

- จงเขียนผังงานแสดงการเขียนและส่งจดหมาย



การพัฒนาอัลกอริธึม

- ตัวอย่าง โครงสร้าง IF

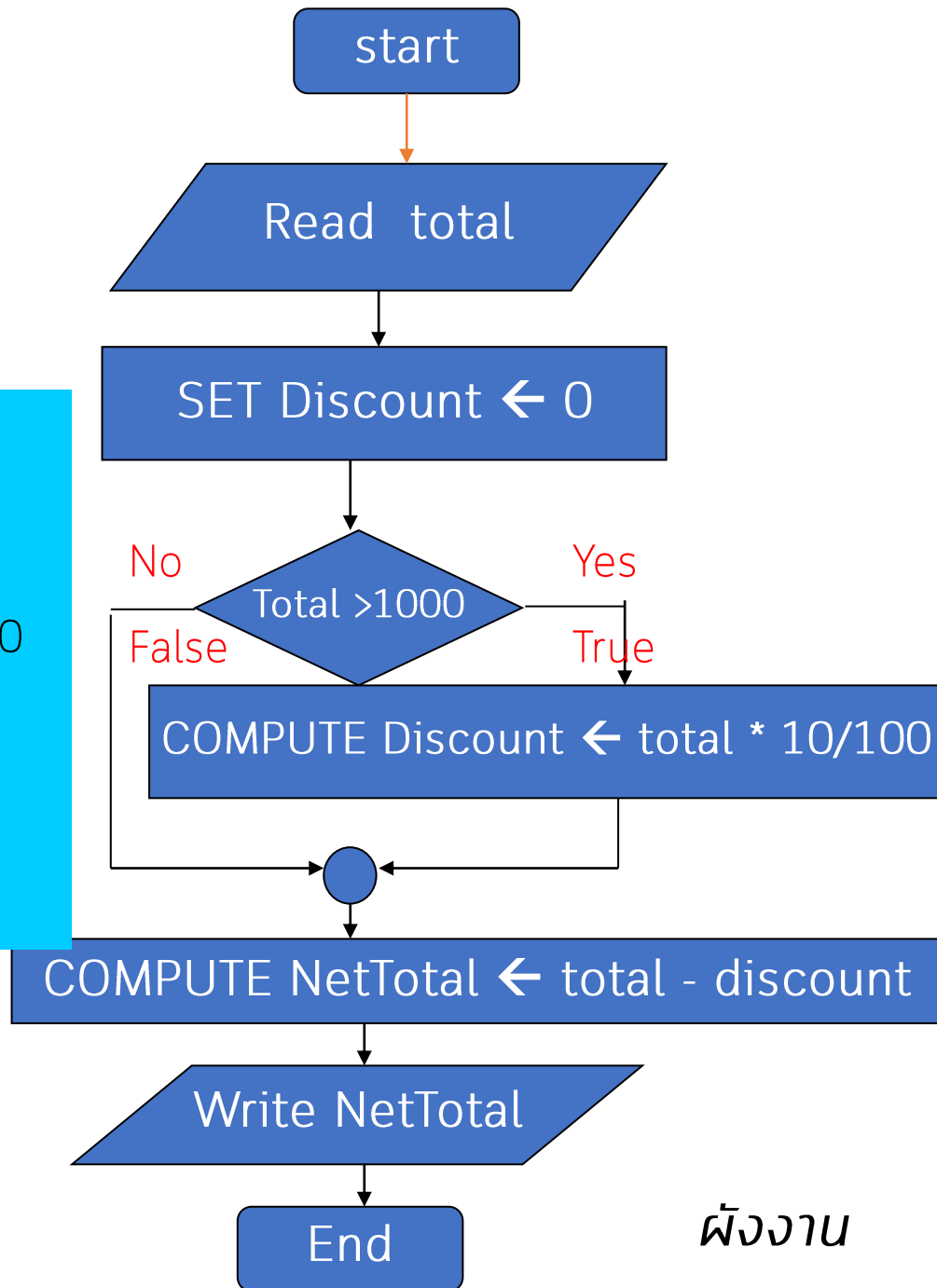
- จงเขียนผังงาน แสดงยอดขาย ถ้าซื้อสินค้ามากกว่า 1000 บาทมีส่วนลดให้ 10%
- วิธีคิด:
 - Output คือยอดขายสุทธิ
 - Input คือ ยอดขายก่อนหักส่วนลด
 - Process มีรายละเอียดเขียนด้วยผังงานได้ในหน้าถัดไป

การพัฒนาอัลกอริธึม

- ตัวอย่าง(ต่อ) โครงสร้าง IF

```
BEGIN
  READ total
  SET discount ← 0
  IF (total > 1000) then
    COMPUTE discount ← total * 10/100
  END IF
  COMPUTE netTotal ← total - discount
  WRITE netTotal
END
```

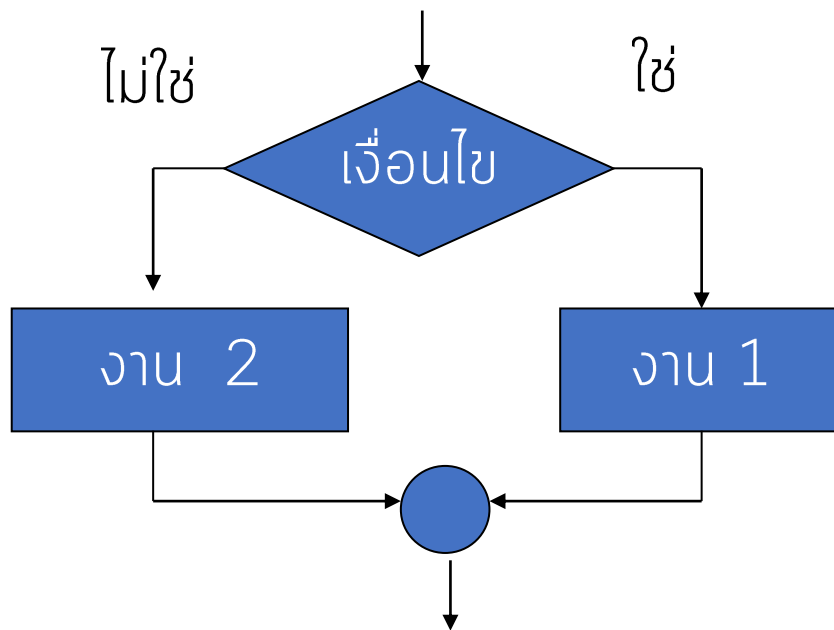
รหัสเทียม



ผังงาน

การพัฒนาอัลกอริธึม

- โครงสร้าง IF-else

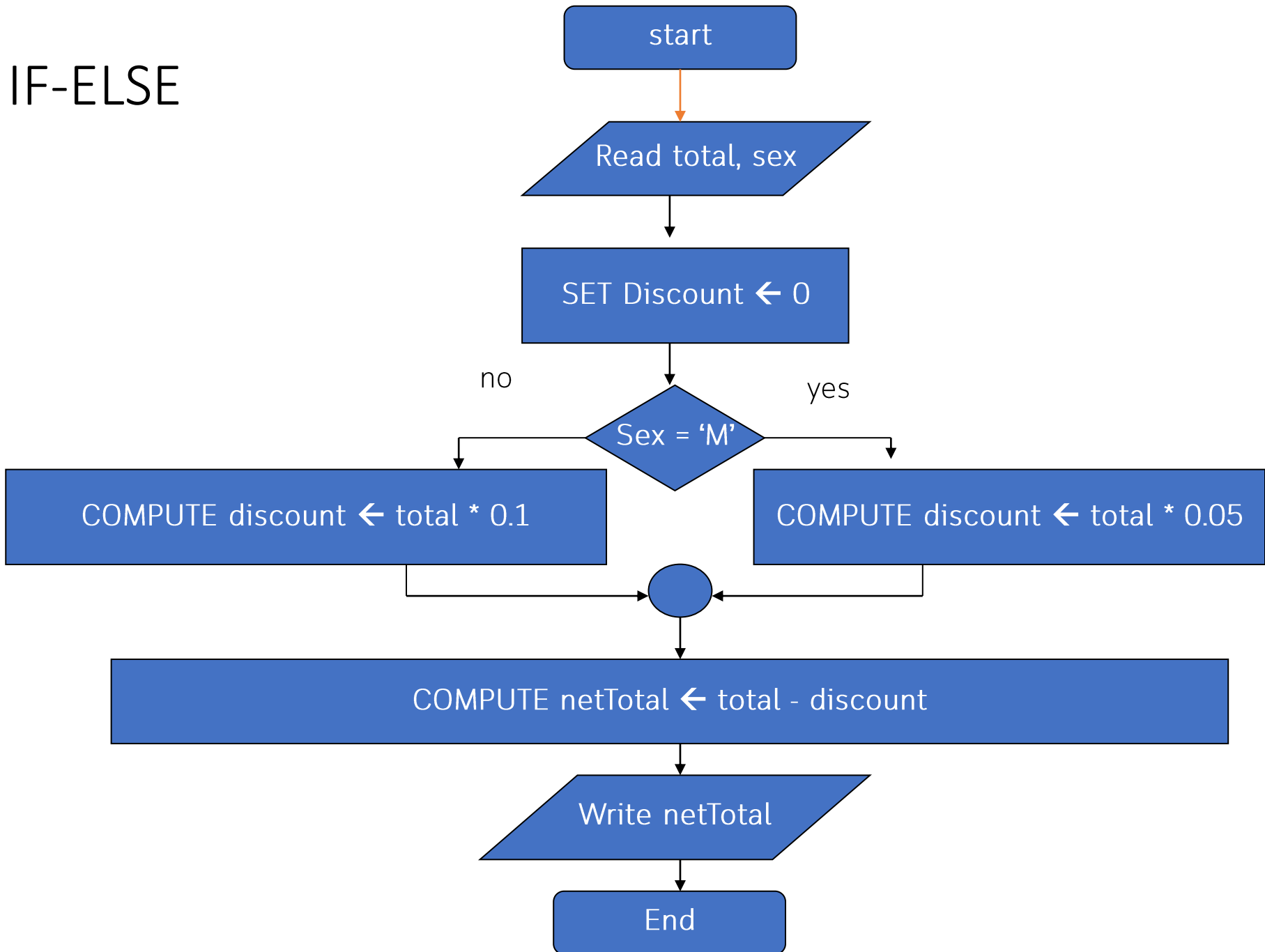


```
If เงื่อนไข  
    งาน 1  
Else  
    งาน 2  
End if
```

การพัฒนาอัลกอริธึม

- โครงสร้าง IF-Else
 - ตัวอย่าง ร้านค้าแห่งหนึ่งมีนโยบาย ลดราคาให้ลูกค้า ถ้าเป็นชายจะลดให้ 5% แต่ถ้าเป็นหญิง จะลดให้ 10% จงเขียนอัลกอริธึมเพื่อพิมพ์ยอดขาย
 - วิเคราะห์ปัญหา
 - Output คือ ราคาสุทธิ กำหนดให้เป็น netTotal
 - Input คือ ราคาสินค้า และ เพศ กำหนดให้เป็น total และ sex ตามลำดับ
 - Process เขียนด้วยผังงาน ดังในหน้าต่อไป

IF-ELSE



$$X = 10$$

$$X = X + 5$$

- $X \leftarrow 10$
- $X \leftarrow X + 5$

โครงสร้างการทำงานแบบมีการเลือก (Selection)

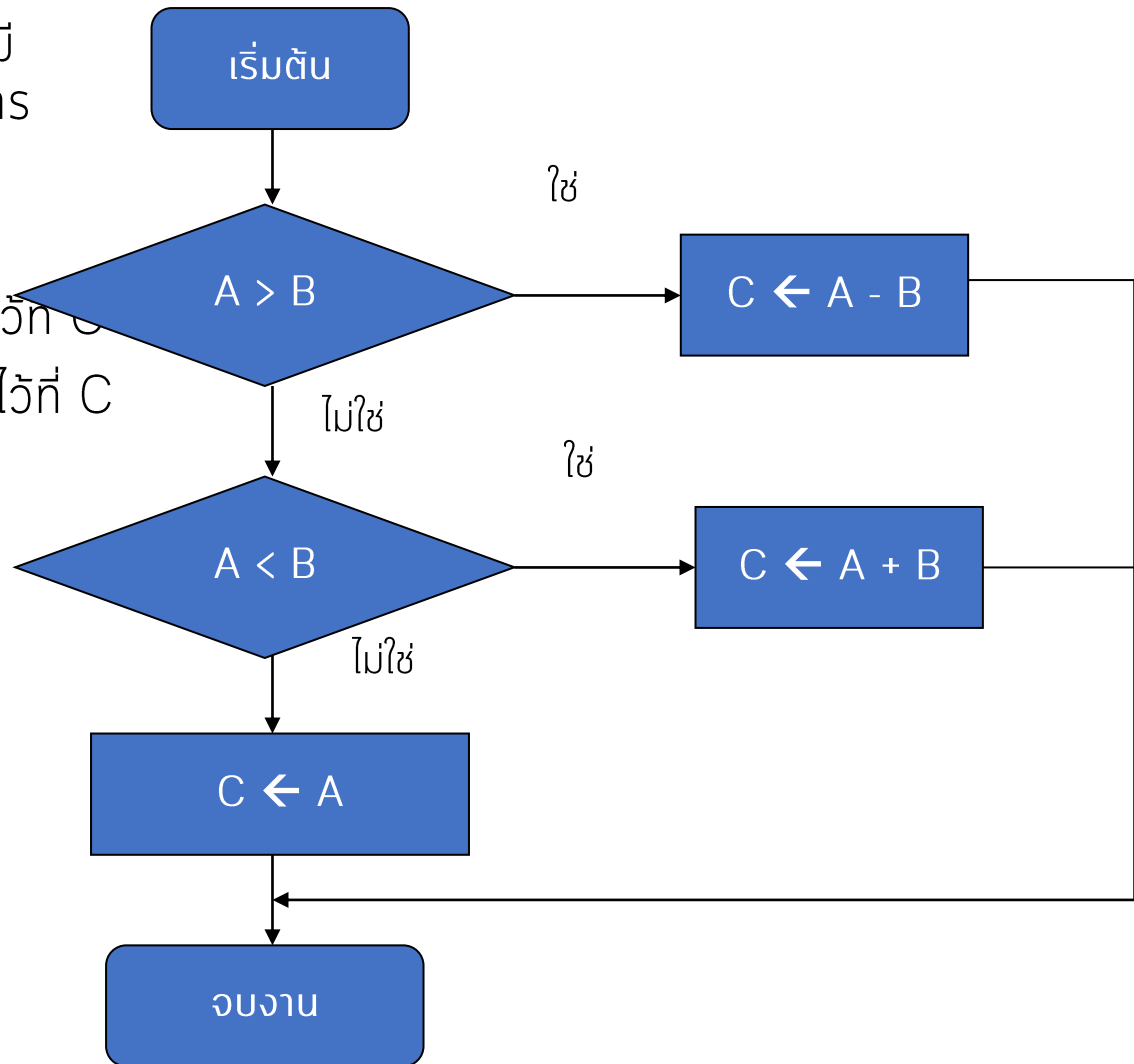
ในบริเวณ A และ B ของหน่วยความจำมีตัวเลขเก็บอยู่ จงเขียนผังงานที่แสดงการเปรียบเทียบระหว่าง A และ B

ตามเงื่อนไข ดังนี้

ถ้า $A > B$ คำนวณ $A - B$ นำผลที่ได้ไปเก็บไว้ที่ C

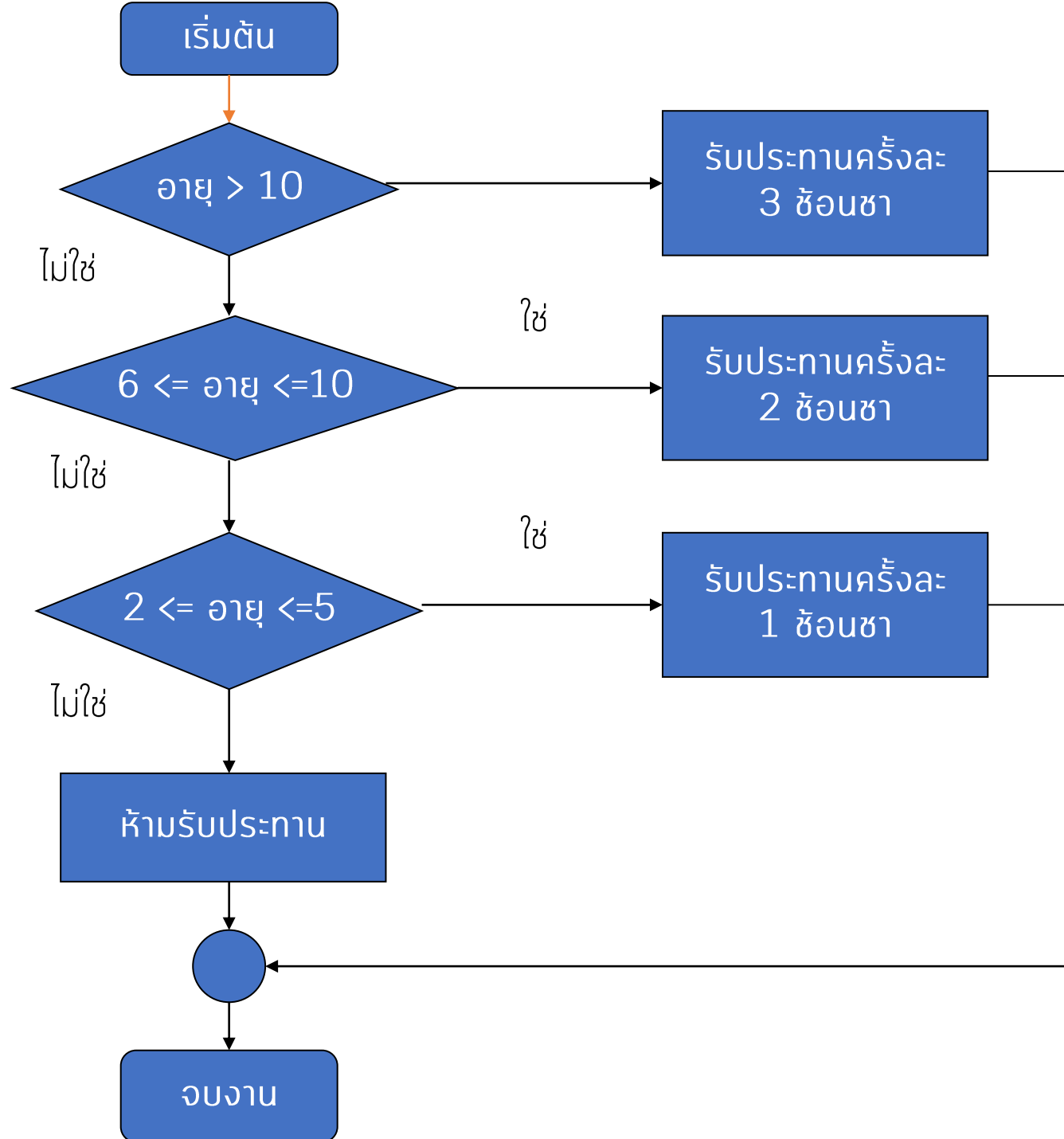
ถ้า $A < B$ คำนวณ $A + B$ นำผลที่ได้ไปเก็บไว้ที่ C

ถ้า $A = B$ เอา ไปเก็บไว้ที่ C



ตัวอย่างการเขียนผังงาน

- เมื่อเราซื้อยาจากร้านขายยาที่มีอยู่ทั่ว ๆ ไปนั้น ในส่วนของวิธีใช้ยามักจะแยกขนาดรับประทานไว้ตามอายุ เช่น
อายุมากกว่า 10 ปี รับประทานครั้งละ 3 ช้อนชา
อายุ 6-10 ปี รับประทานครั้งละ 2 ช้อนชา
อายุ 2-5 ปี รับประทานครั้งละ 1 ช้อนชา
เด็กอายุต่ำกว่า 1 ปี ห้ามรับประทาน
- จงเขียนผังงานที่แสดงถึงขนาดของการใช้ยาตามอายุนี้

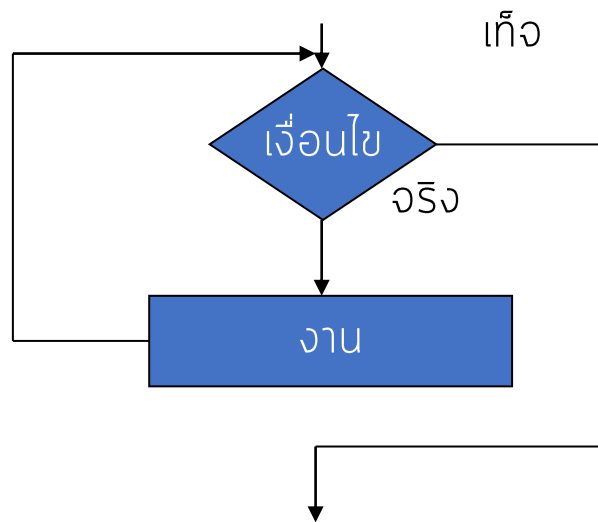


การพัฒนาอัลกอริธึม

- โครงสร้างทำซ้ำ (Repetition Structure)
 - While Repetition Structure
 - โครงสร้างที่ต้องตรวจสอบเงื่อนไขก่อนทำในลูป
 - Do While Repetition Structure
 - โครงสร้างที่ต้องประมวลผลในลูปอย่างน้อย 1 รอบก่อนทดสอบเงื่อนไขออกจากลูป

การพัฒนาอัลกอริธึม

- โครงสร้าง while



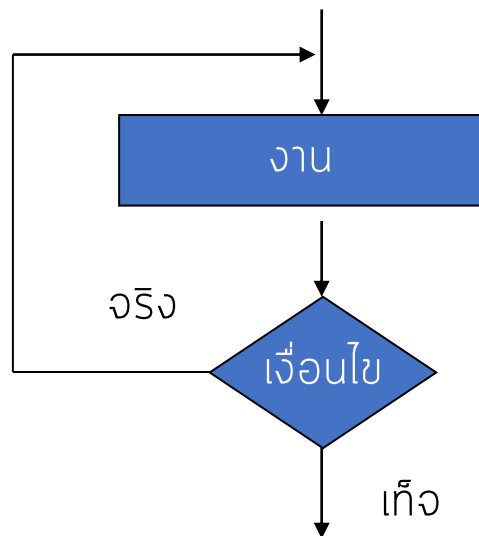
While เงื่อนไข
งาน
end while

ผังงาน

รหัสเทียม

การพัฒนาอัลกอริธึม

□ โครงสร้าง DO while



ผังงาน

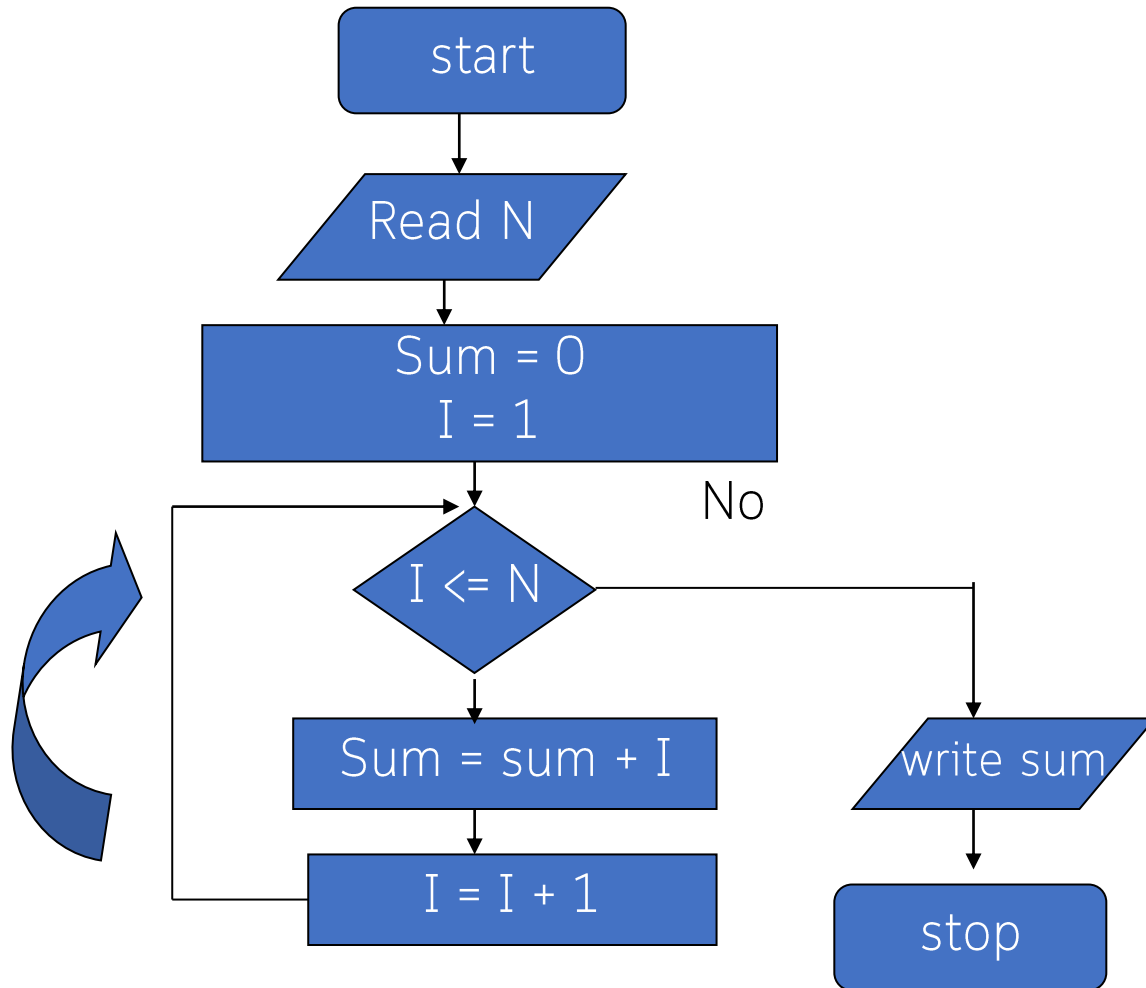
DO
คำสั่งต่าง ๆ
While เงื่อนไข

รหัสเทียม

การพัฒนาอัลกอริธึม

- ตัวอย่าง โครงสร้าง while
- จงเขียน โปรแกรมคำนวณหาผลรวมของสมการต่อไปนี้
$$\text{sum} = 1 + 2 + 3 + \dots + N$$
- วิเคราะห์ปัญหา
 - Output คือ sum
 - Input คือ N
 - Process มี งาน 3 งาน
 1. งาน รับค่า N
 2. งานคำนวณค่าของ sum (วงลูป ด้วยโครงสร้าง while)
 3. พิมพ์ผลลัพธ์ sum

ตัวอย่าง โครงสร้าง while

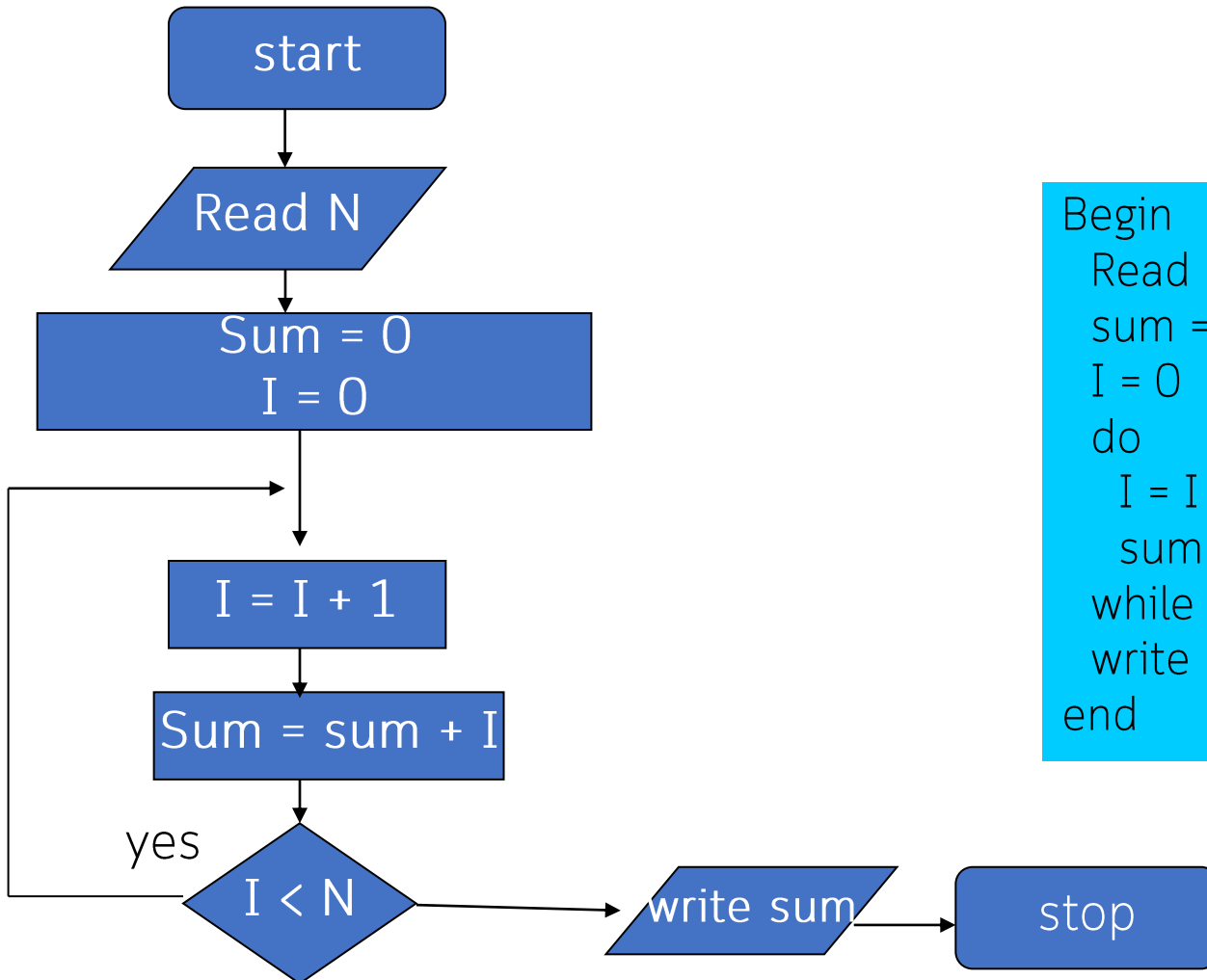


ผังงาน

```
begin
  Read N
  Sum = 0
  I = 1
  While (I <= N)
    sum = sum + I
    I = I + 1
  End while
  Write "sum" , sum
end
```

รหัสเทียม

ตัวอย่าง โครงสร้าง Do while

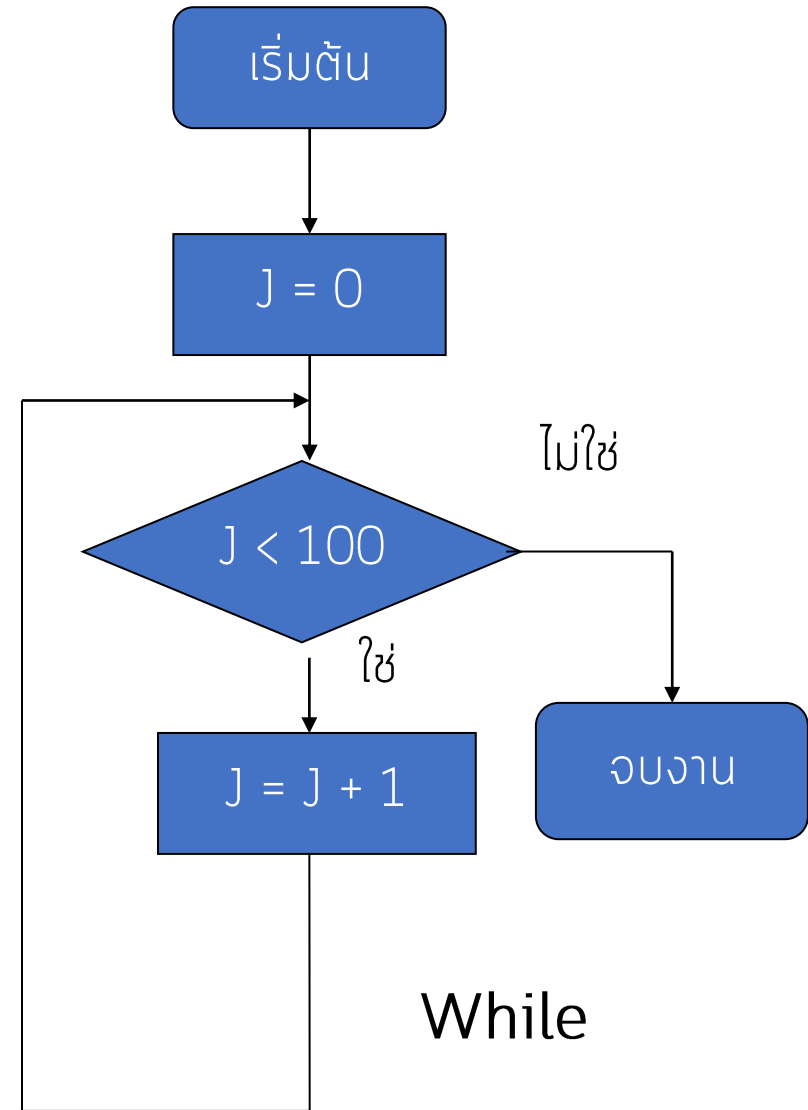
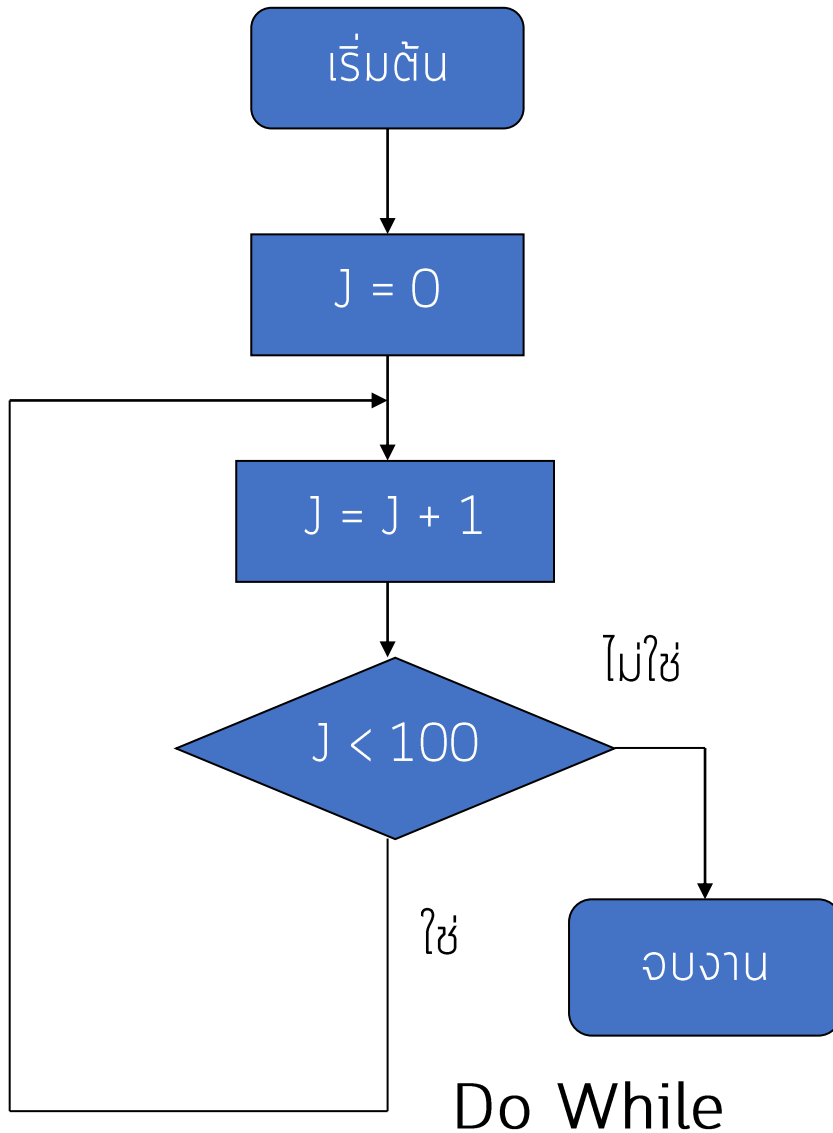


```
Begin  
Read N  
sum = 0  
I = 0  
do  
    I = I + 1  
    sum = sum + I  
while (I < N)  
write Sum  
end
```

ผังงาน

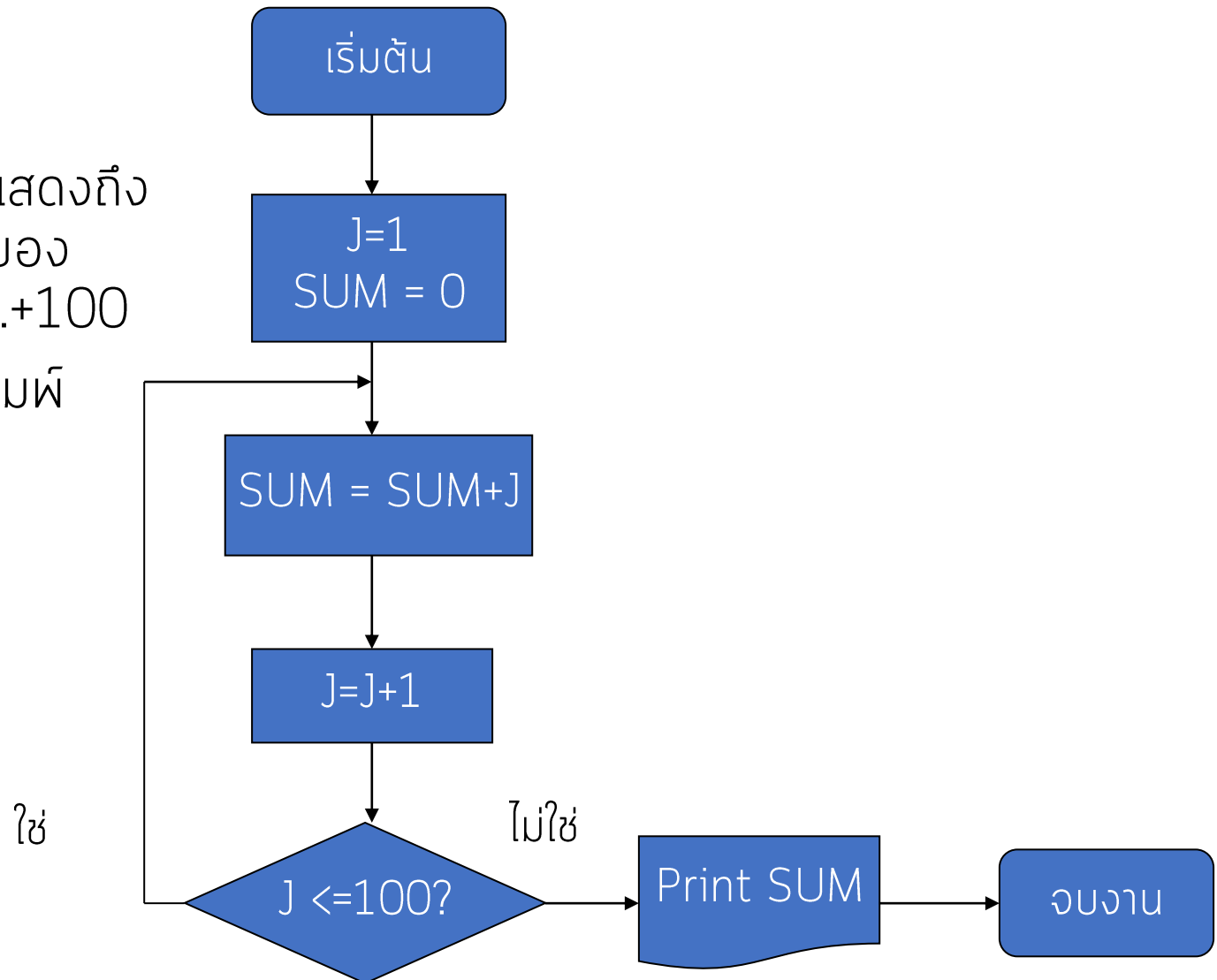
รหัสเทียม

โครงสร้างการทำงานแบบมีการทำงานซ้ำ (Iteration)



โครงสร้างการทำงานแบบมีการทำงานซ้ำ (Iteration)

จงเขียนผังงาน แสดงถึง
การพิมพ์ค่าบวกของ
 $1+2+3+4+5+\dots+100$
ออกทางเครื่องพิมพ์



การเขียนโปรแกรมโดยใช้ภาษาใดภาษาหนึ่ง(Coding)

- การเขียนโปรแกรมที่ดีนั้น ควรจะทำตามขั้นตอน คือเริ่มตั้งแต่ วิเคราะห์ ปัญหา ก่อนแล้วทำการออกแบบโปรแกรม จึงเริ่มเขียนโปรแกรม สำหรับ ผู้ที่ไม่มีประสบการณ์ควรทดลองเขียนในกระดาษก่อน แล้วตรวจสอบ จนแน่ใจว่าสามารถทำงานได้แล้วจึงทำการคีย์ลงเครื่อง

การตรวจสอบข้อผิดพลาดของโปรแกรม (Testing and Debugging the Program)

- รูปแบบข้อผิดพลาด มี 3 แบบคือ
 - Syntax Error - ข้อผิดพลาดจากการใช้ไวยากรณ์ภาษาที่ผิด หรือ อาจเกิดจากการสะกดคำผิด
 - Run-time Error - ข้อผิดพลาดในระหว่างการปฏิบัติงาน(Execution) มักเกิดจากความรู้อันไม่ถึงการณ์
 - Logical Error - ข้อผิดพลาดที่หาและแก้ได้ยากที่สุด ต้องทำการไล่โปรแกรมทีละคำสั่งเพื่อหาข้อผิดพลาดนั้น

การตรวจสอบข้อผิดพลาดของโปรแกรม (Testing and Debugging the Program)

- วิธีการตรวจสอบข้อผิดพลาดของโปรแกรม มีดังนี้
 - ▣ การตรวจสอบด้วยตนเอง (Self Checking) เขียนโปรแกรมลงกระดาษแล้วไล่เช็คตรวจสอบการทำงานทีละขั้นด้วยตนเอง ว่าจะมีการทำงานที่ถูกต้องตามความต้องการหรือไม่
 - ▣ ตรวจสอบด้วยการแปลโปรแกรม (Translating) การแปลเป็นภาษาเครื่อง ตรวจสอบข้อผิดพลาดของโปรแกรม

การทดสอบความถูกต้องของโปรแกรม (Testing and Validating)

- วิธีทดสอบความถูกต้องของข้อมูล มีดังนี้
 - ▣ กรณีที่ข้อมูลถูกต้อง (valid case) ทดสอบโดยใส่ข้อมูลที่ถูกต้องลงไปโปรแกรม เพื่อทดสอบผลลัพธ์ว่าตรงกับที่ต้องการหรือไม่
 - ▣ การใช้ขอบเขตและความถูกต้องของข้อมูล (Range check and Completeness check) เป็นการเช็คขอบเขตข้อมูล
 - ▣ การใช้ความสมเหตุสมผล (Consistency Check)

การทดสอบความถูกต้องของโปรแกรม (Testing and Validating)

- ❑ ข้อมูลที่เป็นตัวเลขและตัวอักษร (Correct No. and Type character check) ตรวจสอบว่าถ้าเป็นฟิลด์ที่เป็นตัวเลขอย่างเดียว เช่น จำนวนเงิน ก็ควรจะป้อนข้อมูลได้เฉพาะตัวเลขเท่านั้น
- ❑ ข้อมูลเป็นไปตามข้อกำหนด (Existence Check) ข้อมูลที่ป้อนต้องเป็นไปตามที่กำหนดไว้แน่นอนแล้วเท่านั้น

แบบฝึกหัด

- จงเขียนขั้นตอนวิธี การคำนวณหาพื้นที่
 - สามเหลี่ยม
 - วงกลม
- เขียนผังงาน (Flowchart)

อ้างอิง

- จงเขียนขั้นตอนวิธี การคำนวณหาพื้นที่
 - สามเหลี่ยม
 - วงกลม
- เขียนผังงาน (Flowchart)