

순서도(Flowchart)

2012.03.12

Na, Eunchong

Mobile Convergence LAB,
Department of Computer Engineering,
Kyung Hee University.

CONTENTS

- ❑ 순서도(Flowchart)란?
- ❑ 순서도 기호 (Symbols)
- ❑ 순서도의 기본 구조
- ❑ 예제
- ❑ 참고 사항

순서도(Flowchart)란? (1/2)

□ 프로그래밍 과정의 단계

- 문제의 이해
- 논리의 설계
- 프로그램의 코딩
- 프로그램의 번역
- 프로그램의 테스트
- 프로그램의 활용

순서도(Flowchart)란? (2/2)

- ❑ 프로그래밍 문제의 해결을 위한 **논리의 설계**
 - 순서도 (flowchart)
 - 의사코드 (pseudocode)
- ❑ 순서도 (flowchart)
 - 어떤 **문제를 해결**하는데 필요한 **논리적인 단계들을 그림**으로 표현한 것
 - **논리적인 흐름**을 나타내는 도구
 - 명령문들의 **연관 관계**를 시각적으로 보여줌
- ❑ 의사코드 (pseudocode)
 - **논리적인 단계들을 실제로 말하는 언어**와 유사하게 나타낸 것

순서도 기호 (Symbols) (1/8)

□ 순서도 작성 방법

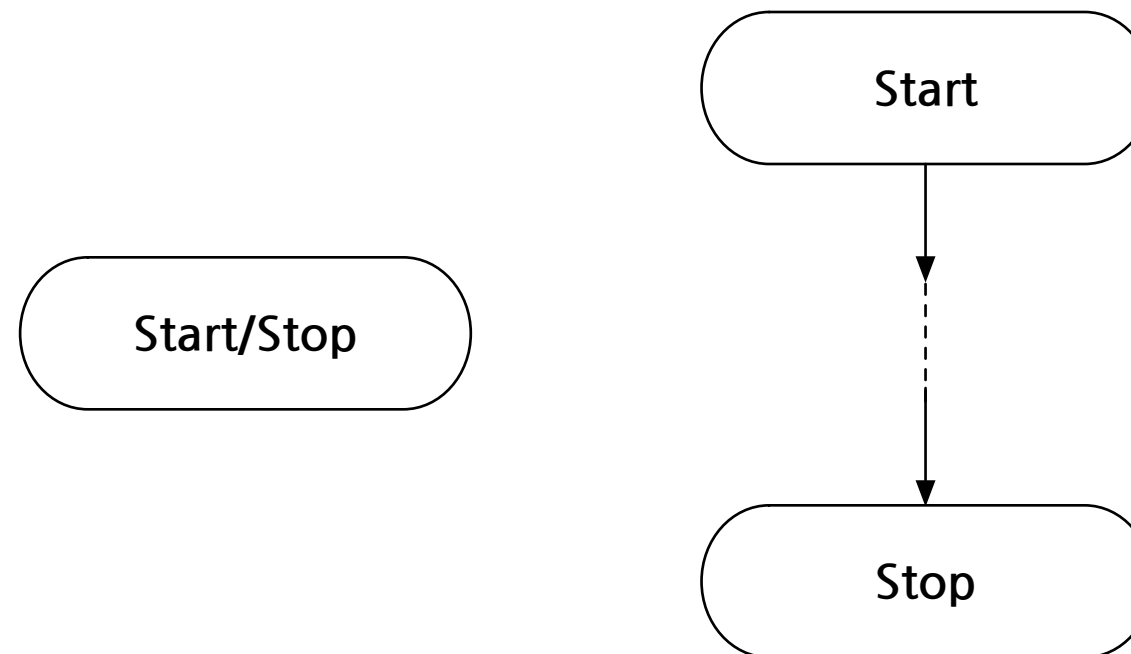
1. 각 명령문의 둘레에 **도형**의 모양을 그림
2. 명령문들의 정확한 순서를 표현하기 위해 **화살표(flowline)**를 이용

□ 대부분의 프로그램은 **입력, 처리, 출력** 단계를 포함

순서도 기호 (Symbols) (2/8)

❑ 터미널 기호 (terminal symbol)

- 순서도의 **시작과 끝**을 나타내는 기호
- 모서리가 둥근 사각형 모양
- 일반적으로 첫 번째 터미널 기호에 **start** 마지막 터미널 기호에 **stop**

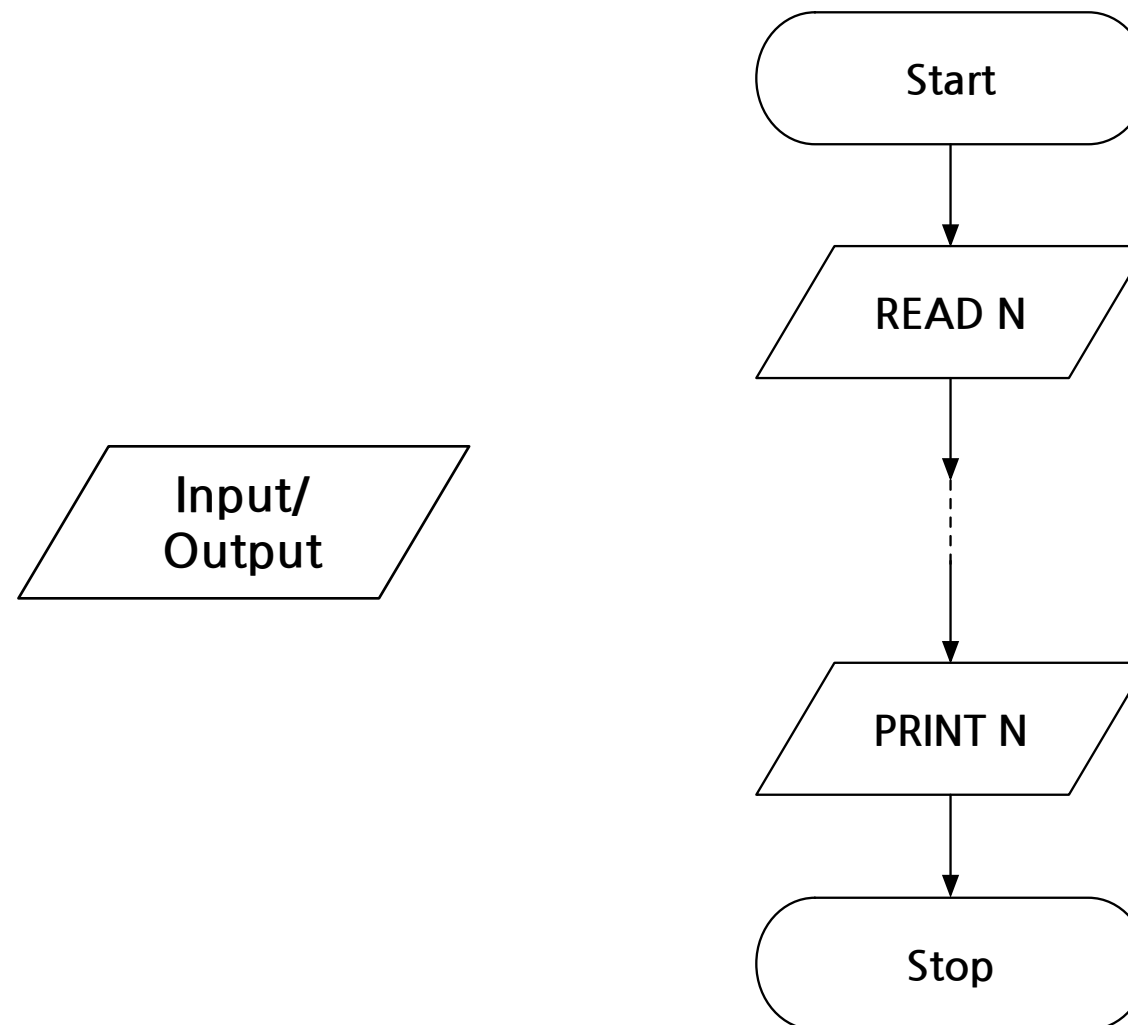


<그림 1> 터미널 기호

순서도 기호 (Symbols) (3/8)

❑ 입/출력 기호 (Input/Output Symbol)

- 입력연산과 출력연산을 나타내는 기호
- 평행사변형

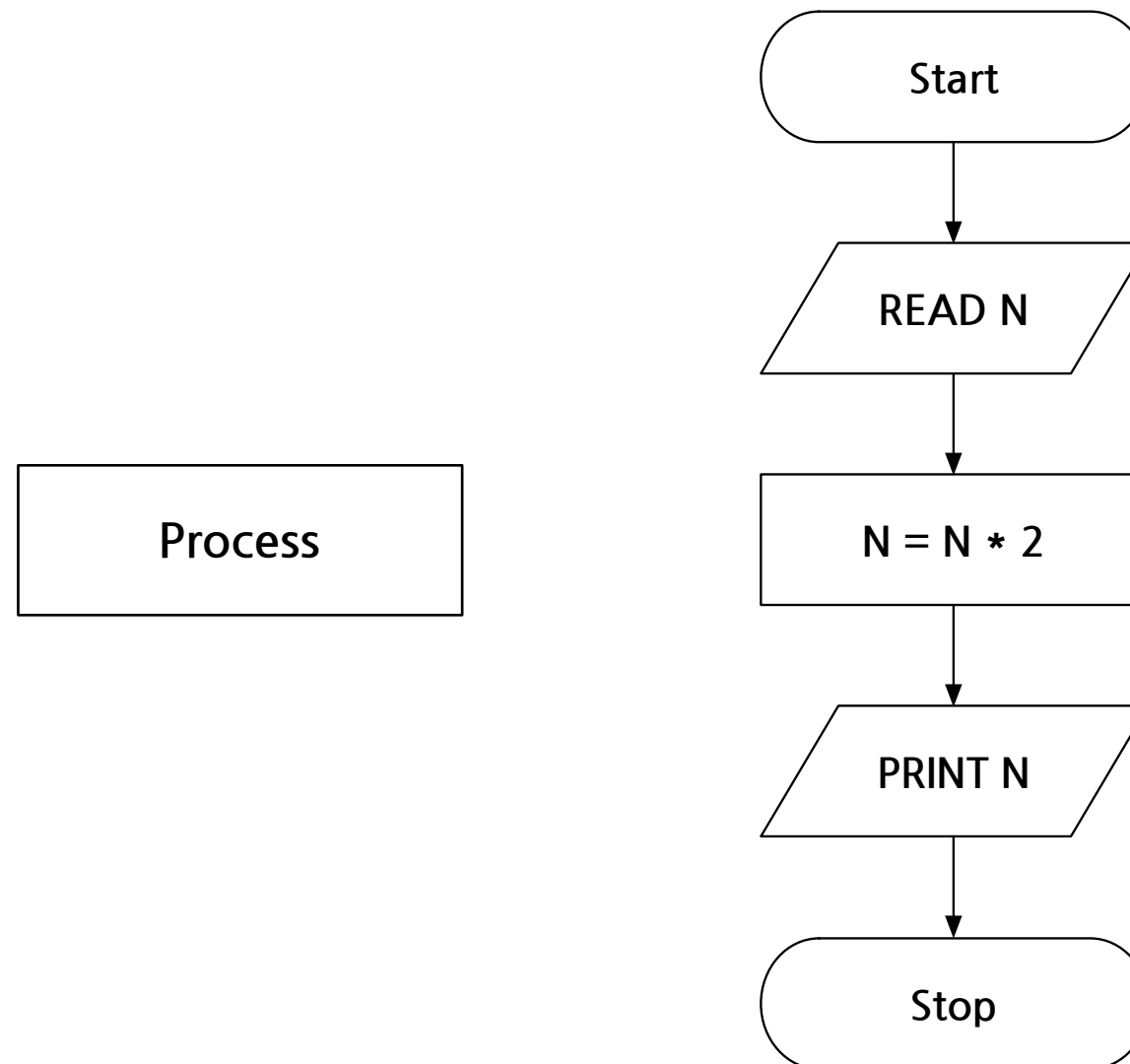


<그림 2> 입/출력 기호

순서도 기호 (Symbols) (4/8)

❑ 처리 기호 (Processing Symbol)

- 수학적 연산 명령문 등 처리해야 할 작업 내용
- 직사각형

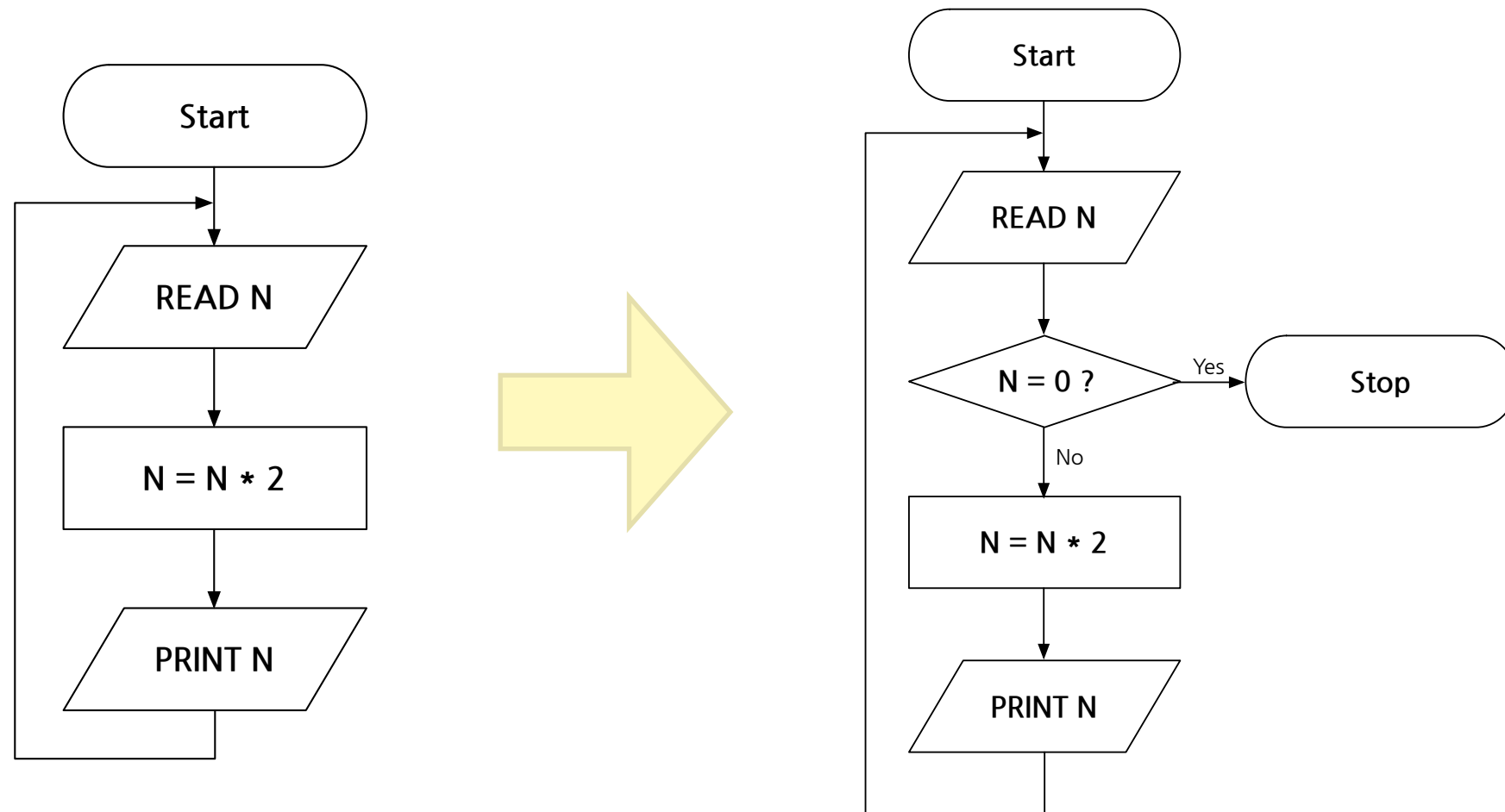


〈그림 3〉 처리 기호

순서도 기호 (Symbols) (5/8)

❑ 판단 기호 (Decision Symbol)

- 순서도에서 어떤 **판단**을 나타내기 위한 기호
- 답이 예(yes)/아니오(no) 중 하나인 질문을 포함
- 다이아몬드

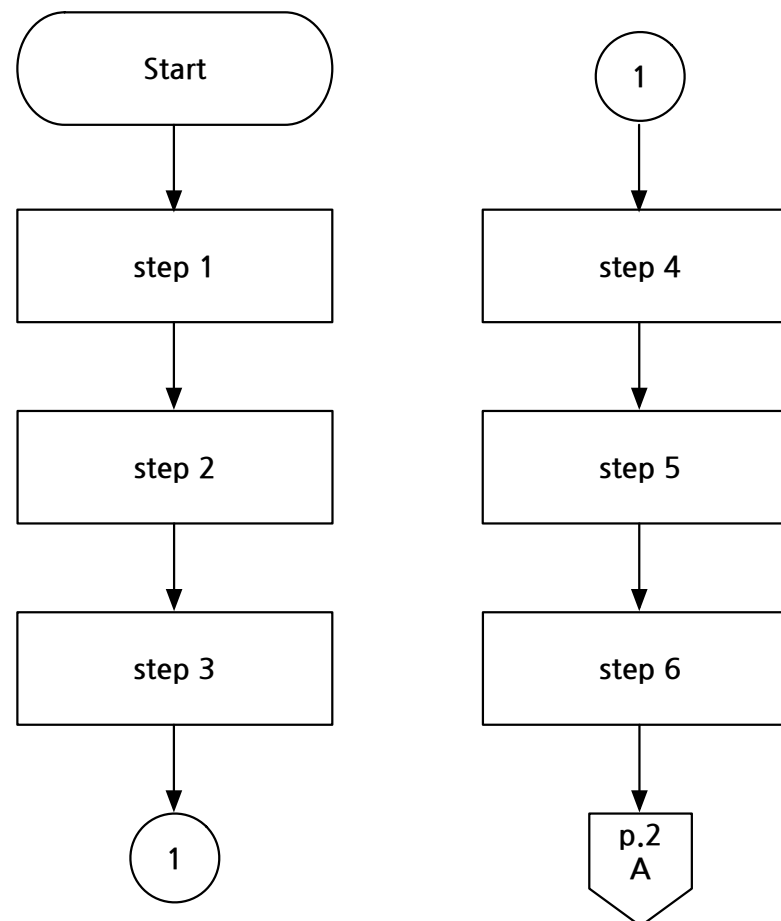


<그림 4> 판단 기호

순서도 기호 (Symbols) (6/8)

❑ 연결자 기호 (Connector Symbol)

- 순서도를 한번에 나타내기 어려울때 **연속해서 그리기 위한 기호**
- 한 페이지 내에서의 연결자 기호 - 원
- 페이지를 넘어가는 연결자 기호 - 아래를 가리키는 오각형

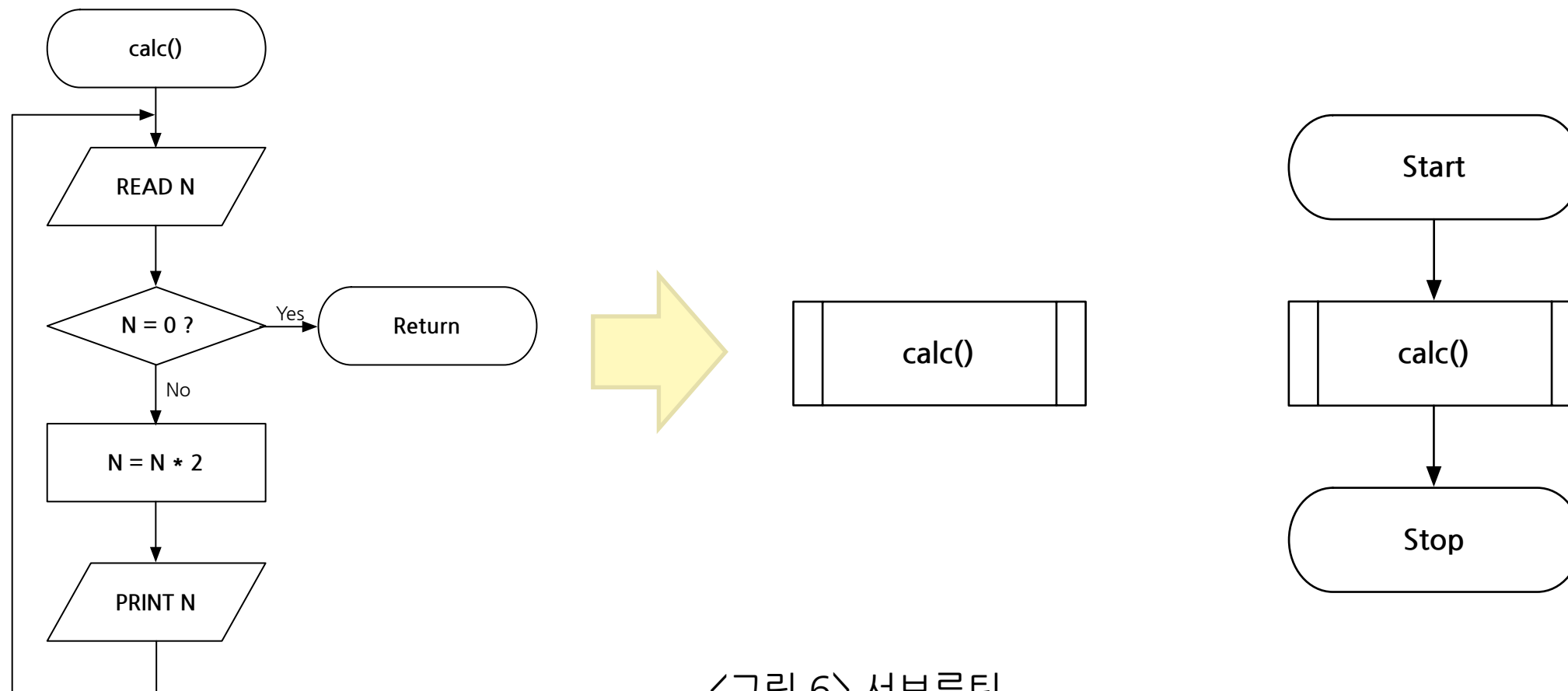


<그림 5> 연결자 기호

순서도 기호 (Symbols) (7/8)

❑ 서브루틴 (Subroutine - Predefined Process)

- 프로그램을 적당한 크기로 나눈 **단위**
 - 모듈(module), 프로시저(procedure), 함수(function), 메소드(method)
- **모듈화(modularization)**의 장점
 - 추상화, 작업의 배분, 재사용 등

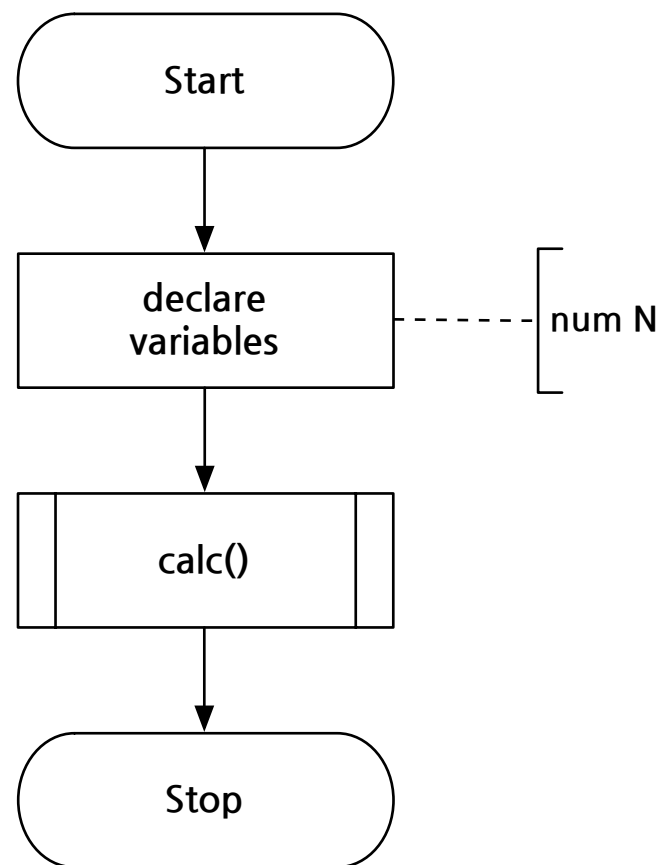


<그림 6> 서브루틴

순서도 기호 (Symbols) (8/8)

□ 주석 (comment)

- 순서도에 **부가적인 설명**을 하기 위해서 사용되는 기호



<그림 7> 주석

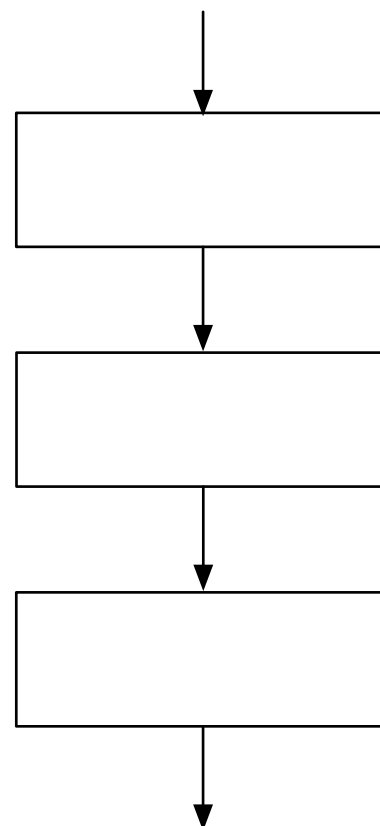
순서도의 기본구조 (1/5)

- ❑ 구조 (structure) : 프로그래밍 **논리**의 **기본적인 단위**
- ❑ 기본적인 3가지 구조를 이용하면 모든 논리를 표현 가능
 - **순서** (sequence)
 - **선택** (selection)
 - **루프** (loop)
- ❑ 한 구조를 다른 구조의 입구 또는 출구에 **연결** 가능
- ❑ 한 구조를 다른 구조의 내부에 **중첩**하는 것이 가능

순서도의 기본구조 (2/5)

□ 순서 구조 (sequence structure)

- 각 동작이 차례대로 수행
- 하나의 순서 구조에는 여러 동작이 포함가능하지만
중간부터 시작하거나 동작을 생략하는 것은 불가능
- 순서를 시작하면 첫 번째 동작부터 마지막 동작까지 차례로 수행

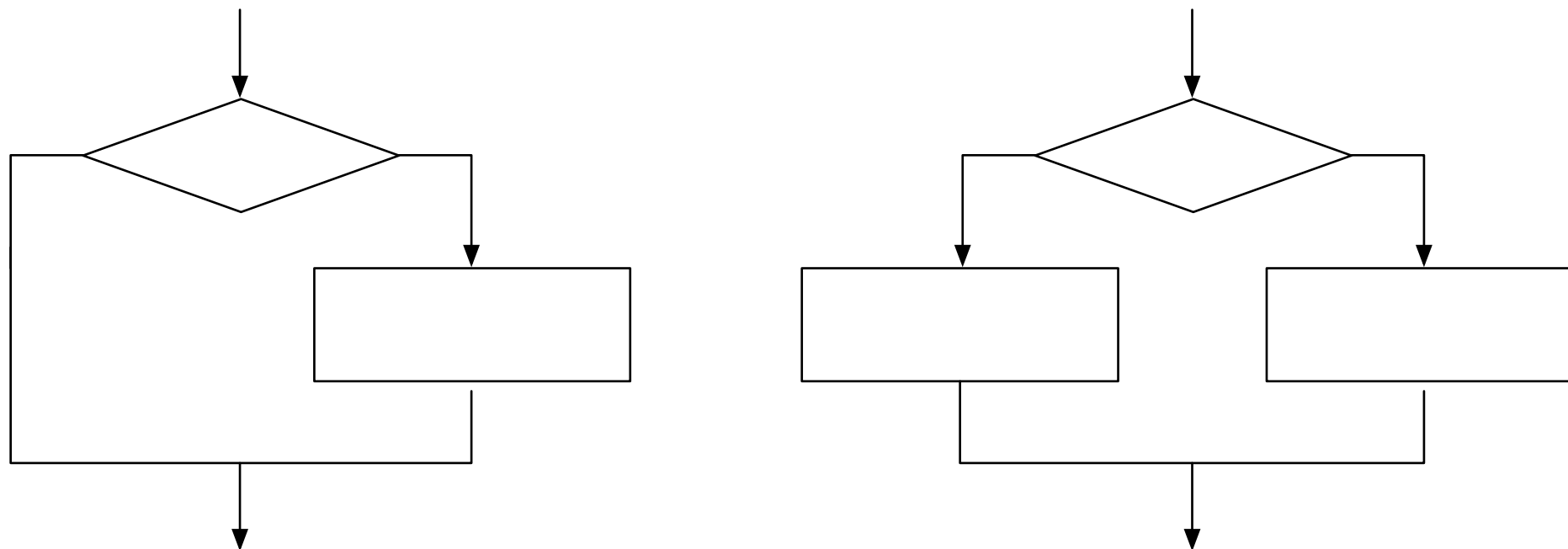


<그림 8> 순서 구조

순서도의 기본구조 (3/5)

□ 선택 구조 (selection structure)

- 판단 구조 (decision structure)
- 질문을 하고 질문의 답에 따라 둘 중 하나의 경로를 따라감
- 단일 대안(single-alternative if) / 이중 대안(dual-alternative if)

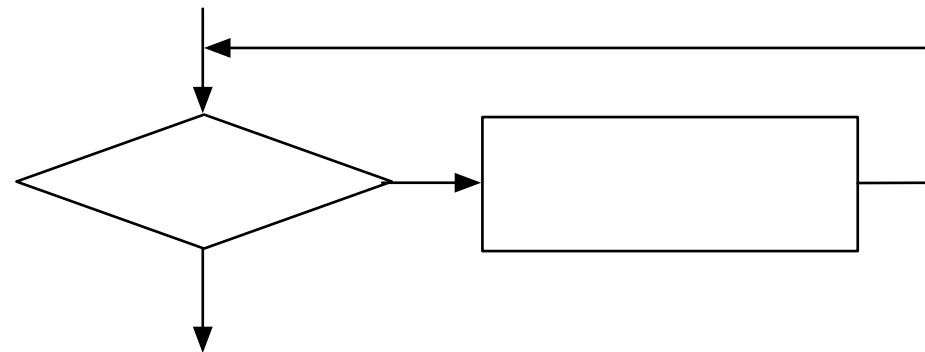


〈그림 9〉 선택 구조

순서도의 기본구조 (4/5)

□ 루프 구조 (loop structure)

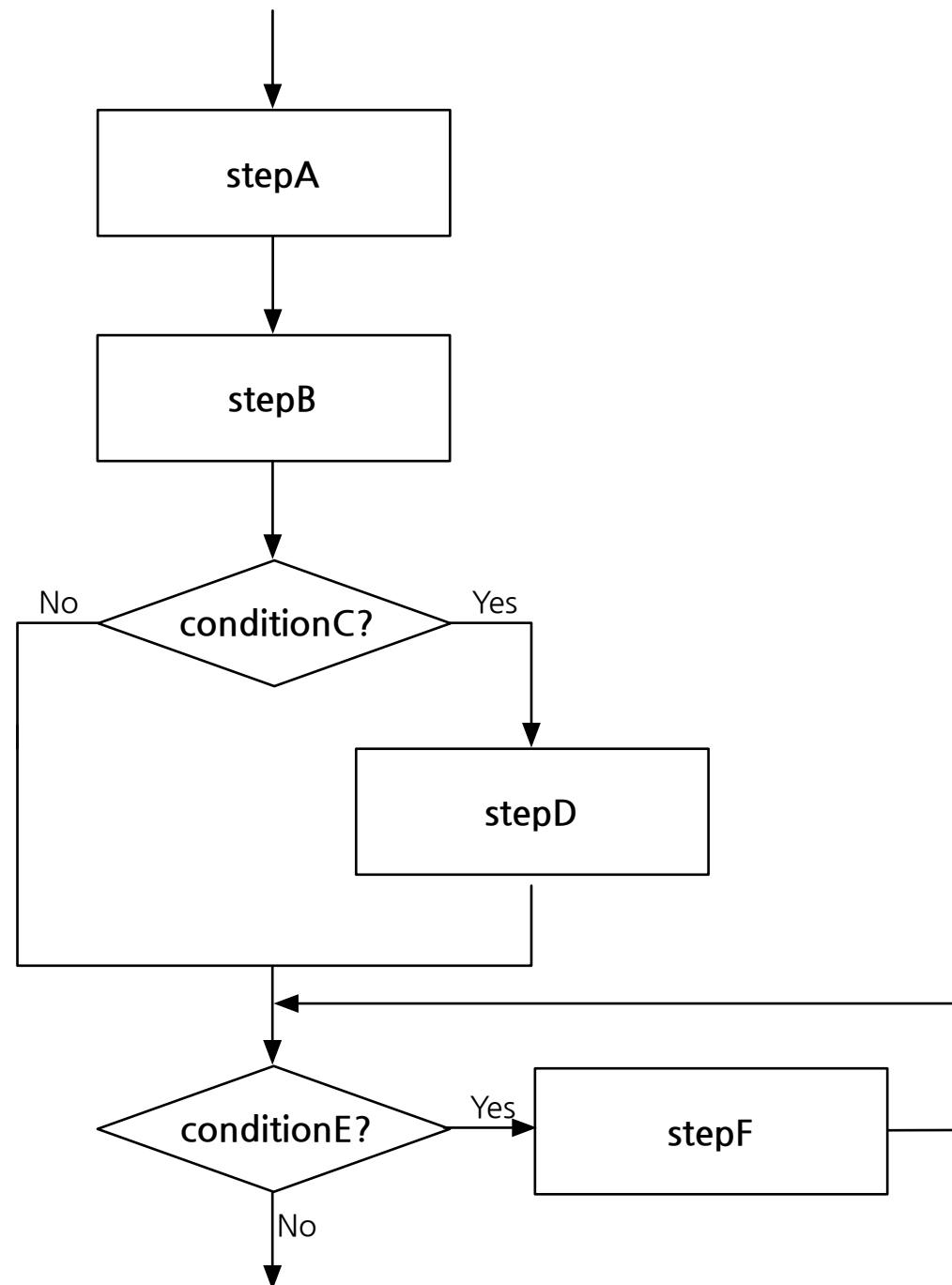
- 판단 구조 (decision structure)
- 질문을 하고 답이 동작을 요구하면 동작 수행 후 다시 질문하는 구조
- 답이 동작을 요구하지 않으면 루프에서 빠져나옴



<그림 10> 루프 구조

순서도의 기본구조 (5/5)

□ 구조적인 순서도와 의사코드



```

do stepA
do stepB
if conditionC is true then
  do stepD
endif
while conditionE is true
  do stepF
end while
  
```

〈그림 11〉 구조적인 순서도와 의사코드

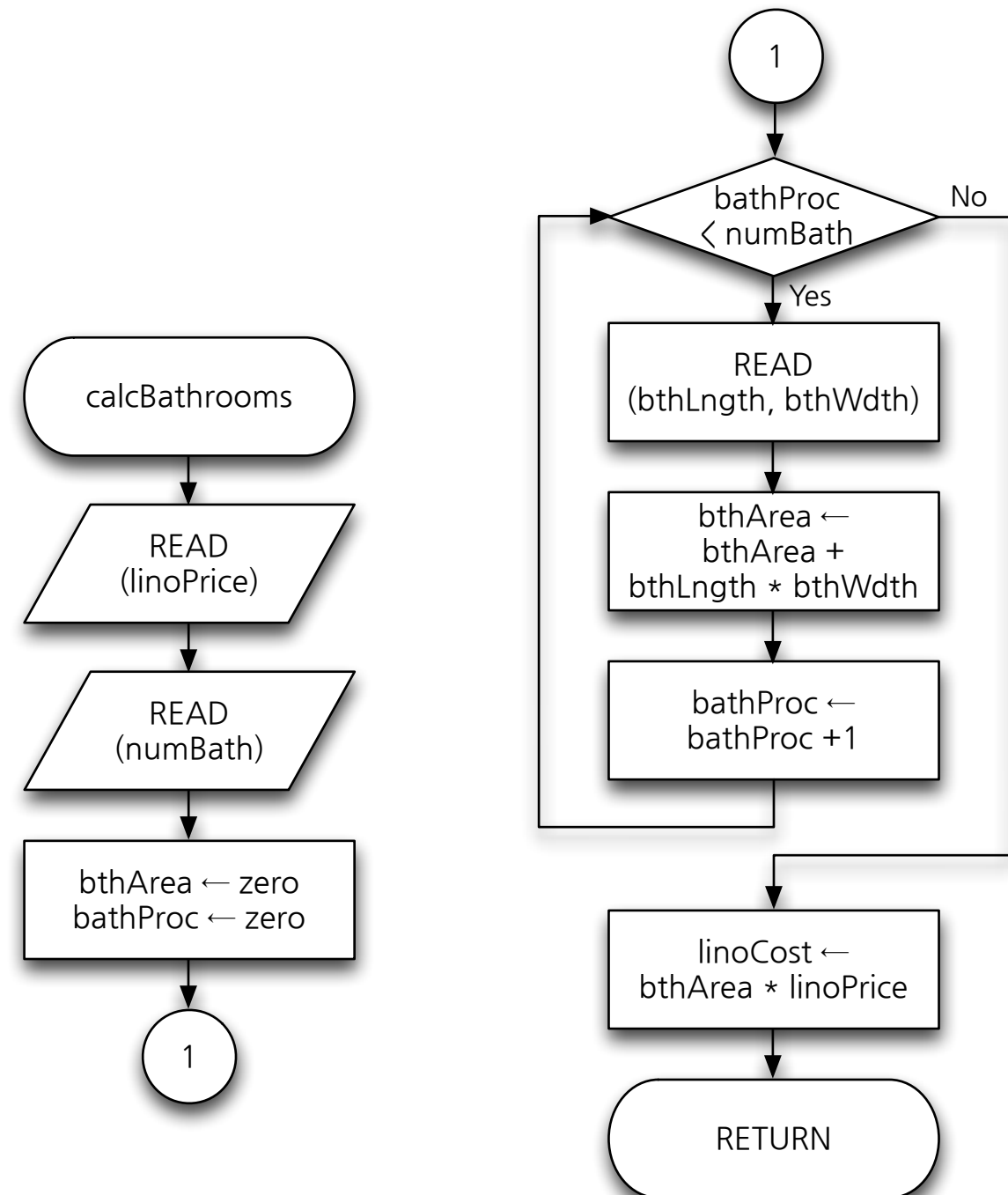
예제 #1

❑ Calculate Bathrooms Problem (p16)

```

Algorithm Calculate Bathrooms
1 prompt user and read linoleum price
2 prompt user and read number of bathrooms
3 set total bath area and baths processed to zero
4 while (baths processed < number of bathrooms)
    1 prompt user and read bath length and width
    2 total bath area =
    3 total bath area + bath length * bath width
    4 add 1 to baths processed
5 end while
6 bath cost = total bath area * linoleum price
7 return bath cost
end Algorithm Calculate Bathrooms
    
```

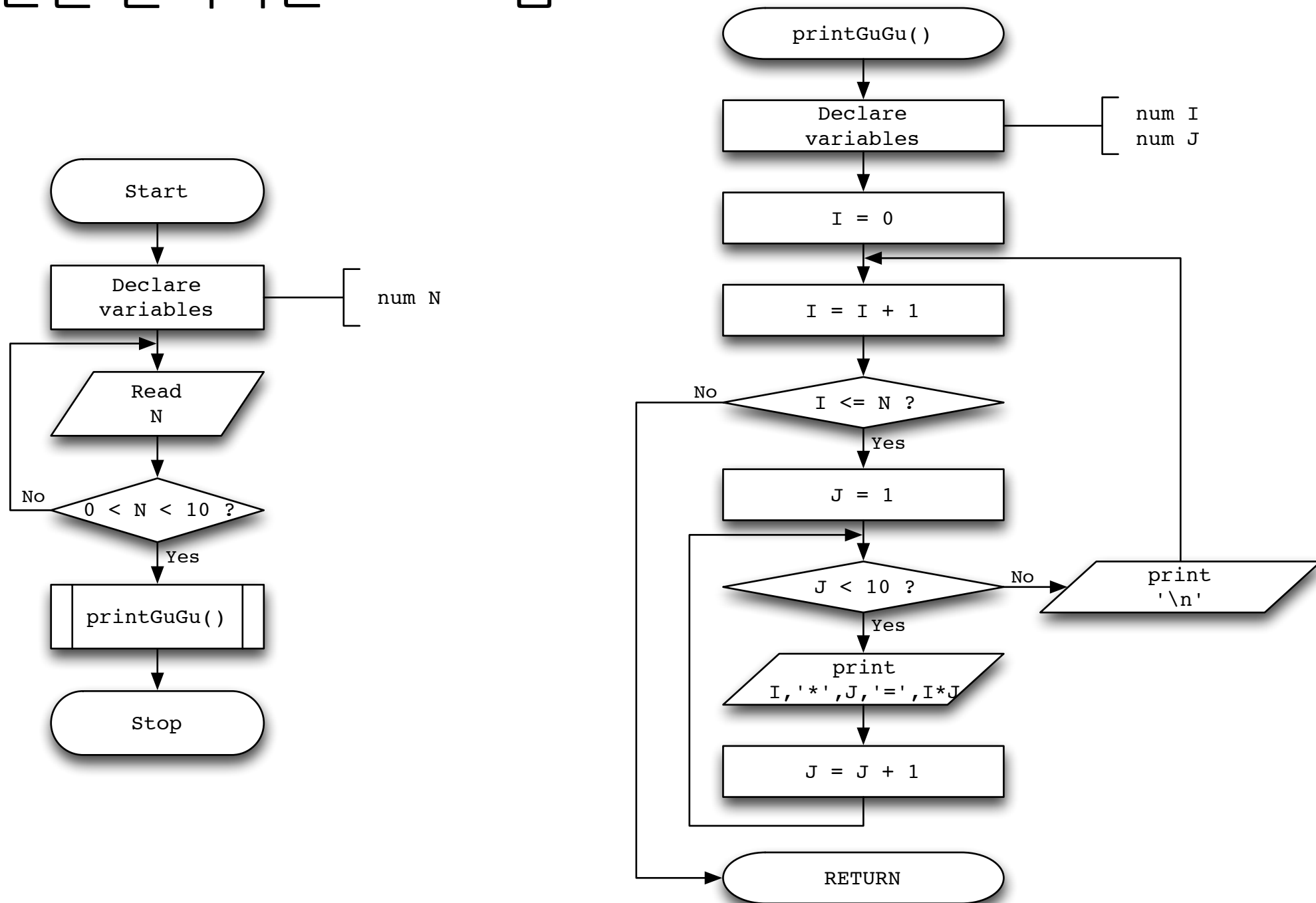
<그림 12> Calculate Bathrooms 문제의 의사코드



<그림 13> Calculate Bathrooms 문제의 순서도

예제 #2

- 문제 정의 : 1~9까지의 정수 n 을 입력받아 1단 부터 n 단 까지 구구단을 출력하는 프로그램



〈그림 14〉 구구단 문제의 순서도

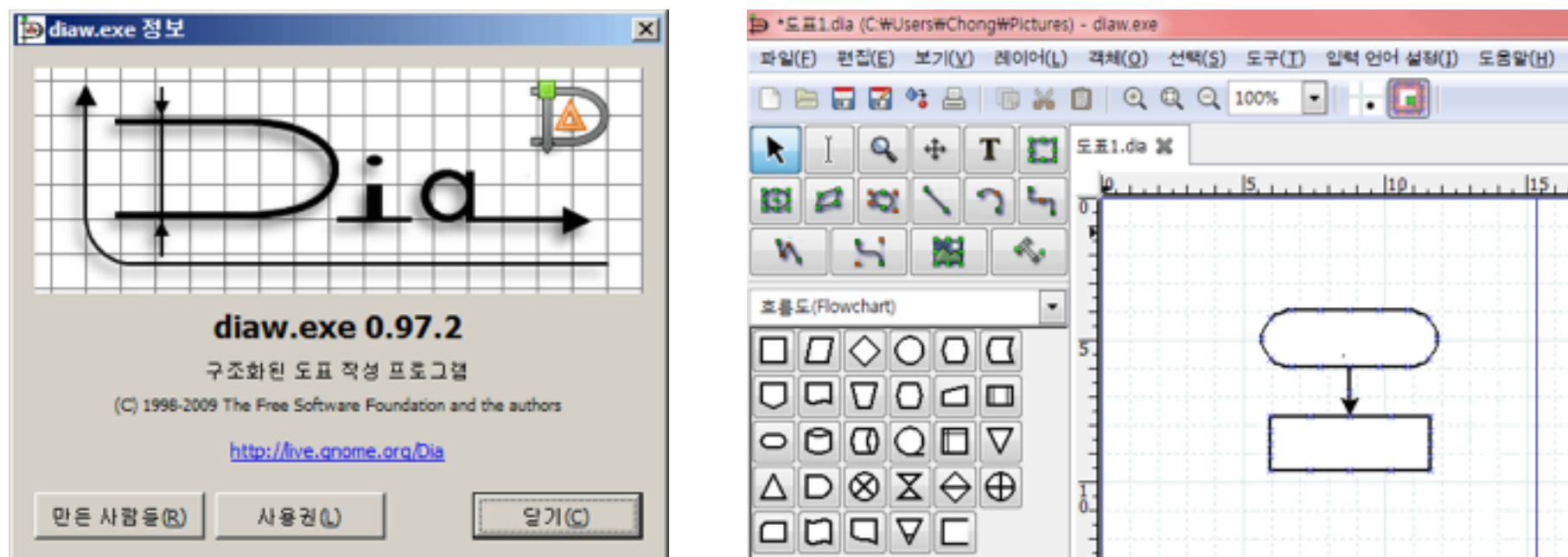
참고 사항

□ 순서도를 그리기 위한 도구

• 윈도우용 프리웨어

■ Dia

- ▶ 순서도, 각종 UML 등 설계 시 유용하게 사용
- ▶ 오픈소스 프로젝트 홈페이지 : <http://live.gnome.org/Dia>
- ▶ 다운로드 홈페이지 : <http://dia-installer.de/>
- ▶ 설치 시 언어 선택을 English로 하면 자동으로 한글 메뉴로 변경됨



〈그림 15〉 다이어그램 작성 프로그램 “Dia”

Q & A