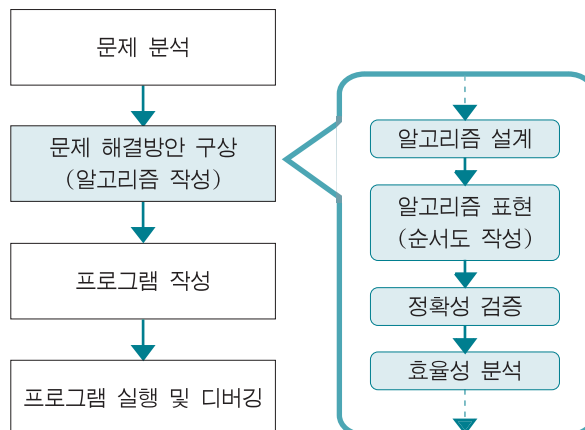


:01 알고리즘(Algorithms) 정의

1. 컴퓨터 프로그램을 통한 문제 해결 절차











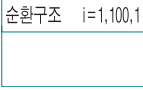
2. 알고리즘(Algorithm)

어떤 문제에 대한 답을 찾는 단계적인 절차

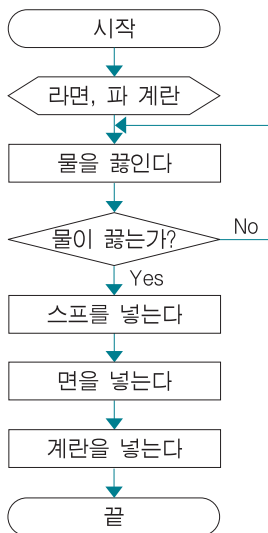
3. 알고리즘 표현 방식

- (1) 자연어(한글, 영어 등) : 표현 및 이해가 쉬우나, 정확한 의미 전달이 어려움
- (2) 알고리즘 기술 언어(=의사코드) : 의사소통에는 문제가 없으나, 프로그래밍 언어에 가까움
- (3) 순서도(Flow Chart)
 - ① 알고리즘을 표준화된 기호로 나타낸 것
 - ② 기호를 사용하므로 의사소통이 명확함
 - ③ 특정 프로그래밍 언어와 독립적이므로 순서도를 배우거나 작성하고 이해하기가 쉬움

:02 주요 순서도 기호

기 호	이 름	의 미
	단자 (Terminal)	순서도의 시작과 끝을 표시
	준비 (Preparation)	변수의 선언 및 초기값 부여, 배열 선언
	처리 (Process)	값을 계산하거나 대입하는 기호
	판단 (Decision)	참과 거짓을 판단하거나 조건에 맞는 경로로 분기
	입·출력(Input/Output)	데이터의 입력과 출력
	문서 (Document)	처리된 결과를 프린터로 출력
	흐름선 (Flow Line)	각종 처리 기호의 처리 흐름을 연결
	연결자 (Connector)	다른 곳으로의 연결을 표시
	순환구조(Loop)	우측 상단의 i=1, 100, 1은 i=초기화, 최종값, 증가치를 의미. 즉, 1부터 100까지 1씩 증가하라는 의미

※ 순서도의 예 (라면 끓이기)



:03 변수(變數) (Vriables)

1. 변수의 정의

- (1) 값 또는 수식의 결과를 저장하는 공간
- (2) 변할 수 있는 값을 의미

2. 변수의 사용 예

예	의 미
$A = 5$	변수 A에 값 5를 저장
$A = 10 + 5$	변수 A에 수식 "10+5"의 계산 결과를 저장 (15가 저장됨)
$A = B + 10$	변수 A에 수식 "B+10"의 계산 결과를 저장
$A = A + 20$	변수 A에 수식 "A+20"의 계산 결과를 저장20

:04 연산자 (Operators)

1. 연산자의 정의

변수나 값의 연산을 위해 사용되는 부호

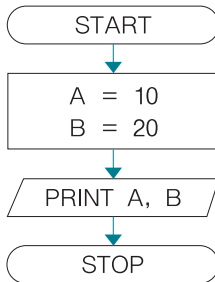
2. 연산자의 종류

산술 연산자		관계 연산자	
연산자	의미	연산자	연산자
\wedge	거듭제곱	$=$	같다
\times	곱하기	$\langle \rangle$	같지 않다
$/$	나누기	\rangle	크다
$+$	더하기	$\rangle =$	크거나 같다
$-$	빼기	\langle	작다
		$\langle =$	작거나 같다

:05 순서도 기본 모형

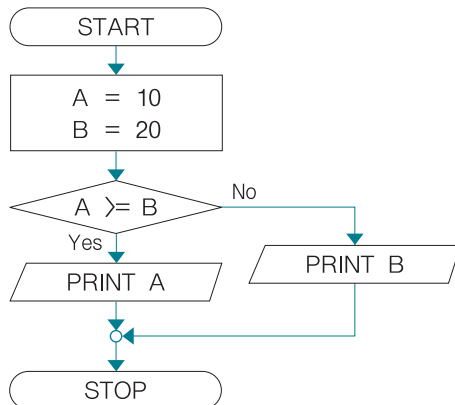
1. 직선형

처음 시작부터 마지막 종료 명령까지 단계적으로 진행되는 순서



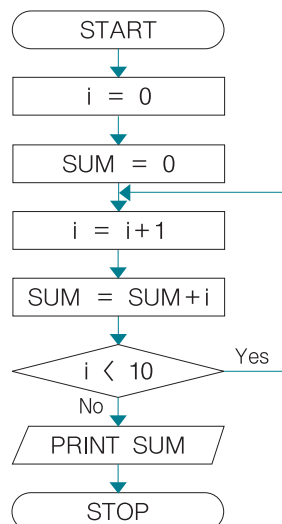
2. 분기형

조건에 따라 실행내용이나 순서를 달리하는 형태



3. 반복형

조건을 만족할 때까지 일정한 내용을 반복해서 수행하는 형태



각 항 (i)	0	1	2	3	4	5	6	7	8	9	10
합계(SUM)	0	1	3	6	10	15	21	28	36	45	55

:06 시험장에서의 알고리즘 해결 공식

1. 문제 발생

시험장에서 문제와 함께 사용되는 변수까지 주어짐

2. 문제 분석

변수와 변수의 공식 \Rightarrow 규칙을 이끌어 냄

3. 알고리즘 작성

시험장에서 순서도로 문제가 주어지며, 5개 또는 6개의 괄호가 비워져 있음

4. 디버깅

(1) 디버깅(Debugging)

컴퓨터에서 발생한 오류를 잡기 위해 소스 코드를 한 줄 한 줄 추적해 보며, 변수 값의 변화를 검사하면서 프로그램이 정확한지, 어디가 잘못되었는지 검증하는 작업

(2) 프로그램으로 디버깅을 할 수 없으므로 수기(手記)로 디버깅하면서 검증을 함

(3) 순서도에 사용된 모든 변수와 조건문의 조건식들을 직접 적어보고, 하나하나 추적해 가며 칸을 채우면서 작성된 순서도가 정확한지 검증해야 함

(4) 디버깅을 거치지 않은 순서도는 정확하다고 말할 수 없음

(디버깅은 알고리즘 만점을 위한 필수 단계)

※ 예제

(1) 문제 발생

정사각형의 넓이를 구하라.

(2) 문제 분석

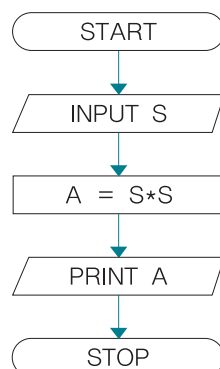
① 정사각형의 한 변의 길이를 변수 S라고 하자.

② 정사각형의 넓이를 변수 A라고 하자.

③ $A = S * S$

(3) 알고리즘 작성

분석 결과를 순서도(Flow Chart)를 이용해 논리적으로 표현한다.



(4) 디버깅

S		A		PRINT	
순서	값	순서	값	순서	값
①	4	②	16	③	16
①	8	②	64	③	64
①	10	②	100	③	100