



1장 순서도

- 1.1 순서도의 개요
- 1.2 순서도의 기본형
- 1.3 순서도에 이용되는 기호
- 1.4 순서도 작성 요령
- 1.5 순서도 프로그램



1.1 순서도의 개요_1

● 순서도의 개념

- 현대의 컴퓨터 프로그래머는 대개 다음과 같은 생각을 가지는 것이 공통적인 의견
- 컴퓨터라는 것은 내가 원하는 것을 수행할 수 있는 것이 아니라, 내가 지시한 것을 수행하는 기계
- 어떤 문제를 해결키 위해, 우리는 컴퓨터에 처리를 하기 위하여 수행되어지는 것을 순서대로 지시를 해야함
- 알고리즘이란 말은 어떤 문제를 해결하려고 할 때 존재하는 규칙들을 나열시켜 수학에서의 용어를 컴퓨터의 용어
- 알고리즘(algorithm)의 어원은 중세 아라비아의 수학자인 Al-Khowrizm이라는 사람의 이름에서 비롯
- 알고리즘은 프로그램 작성을 위한 예비 단계로서 주어진 문제를 해결하기 위한 작업절차 또는 계산절차를 뜻함



1.1 순서도의 개요_2

- 알고리즘의 실제 표현 방법
 - 일상 생활에서 사용하는 언어로 기술하는 방법
 - 특정한 고급 언어의 표기법을 따르는 방법
 - 순서도(flowchart)와 같이 단계적 계산절차를 그림과 기호로 나타내는 방법
- 알고리즘에서 사용하는 명령들
 - 항상 정확하고 간결해야 함
 - 유한 번의 단계를 거쳐 반드시 끝이 나야 함
 - 일반적인 데이터의 입력에 대해서도 적용이 되어야 함
 - 컴퓨터의 계산 효율과 기억용량의 측면에서 효율적이어야 함



1.1 순서도의 개요_3

- 알고리즘은 일반적으로 항상 작성이 되어짐
- 변수(variable)들로 나타내므로 해서 같은 형태의 많은 다른 문제들에서도 쉽게 해결 가능
- 컴퓨터 프로그래밍에 대해, 한 변수는 기억장소의 이름으로서 간주되어질 수가 있음
- 변수는 한순간에 하나의 값을 가질 수가 있음
- 각 컴퓨터 언어에서 변수명을 작성하는데는 규칙들이 존재
- 일반적으로 변수명은 한 문자로 되어질 필요는 없음
- 변수명이라는 것은 프로그래머에게 변수에 저장되어 있는 값을 알려주기 위해서 단어나 약자로 사용하는 것이 일반적
- 알고리즘을 작성한다는 것은 주로 사용되어지는 컴퓨터 프로그래밍 언어(programming language)와는 별개의 일
- 하향식 설계(top-down design)의 기법은 어떤 문제를 이전에 작성할 문제보다 작은 형태로 여러 개의 문제로 분할시키기 위해서 사용하며 분할된 문제들 각각은 더 분할시킬 수 있으면 다시 그보다 작은 형태의 문제로 분할시킴. 상향식(bottom-up)기법은 반대.



1.1 순서도의 개요_4

- 어떤 정보가 유용한지 그리고 어떤 정보를 찾으려고 하는지를 아는 것은 상당히 중요한 것
- 알고리즘을 가끔 입력(즉, 문제 해결을 위해 요구되는 자료를 얻는 것)에서 시작하여 출력(해답을 표시)으로서 끝이 나게 됨
- 알고리즘을 기술하기 위해서는 컴퓨터가 어떤 일을 수행하여 사용자가 원하는 결과를 얻을 수 있도록 단계를 분명하게 기술
- 알고리즘을 기술하는 방법으로는 순서도, 자연어, 프로그래밍언어, 의사언어(pseudo language) 등이 있음
- 의사언어는 그 나라 언어(자연어)와 프로그래밍 언어를 혼합하여 코딩하는 사람이 알기 쉽게 알고리즘을 기술하는 언어

1.1 순서도의 개요_5

● 두 변수들 간의 값을 바꾸는 예

① FIRST와 SECOND의 값을 얻음



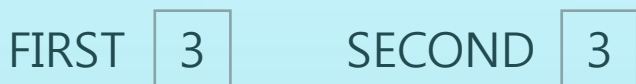
② FIRST에 SECOND의 값을 할당



③ SECOND에 FIRST의 값을 할당



④ FIRST와 SECOND의 값을 나타냄



- 알고리즘에 의해서 값들이 교환 되어지지 않았음
- 값들을 서로 교환하기 위해서, 값들 중 한 값을 저장시키기 위하여 임시 저장장소를 만들기 위해 하나의 변수를 더 만들어야 함

1.1 순서도의 개요_6

- 두 변수들 간의 값을 바꾸는 예(추가 변수 이용)

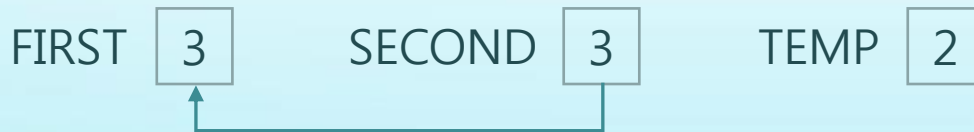
- ① FIRST와 SECOND의 값을 얻음



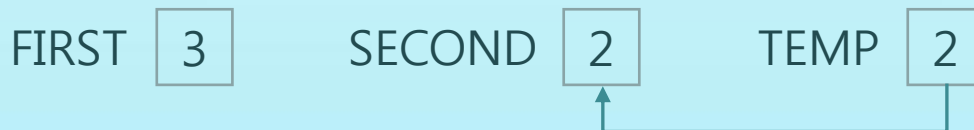
- ② FIRST의 값을 임의의 장소 TEMP에 할당



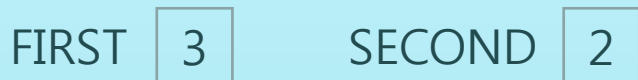
- ③ FIRST에 SECOND 값을 할당



- ④ SECOND에 TEMP의 값을 할당



- ⑤ FIRST와 SECOND의 값을 출력





1.1 순서도의 개요_7

- 어떤 일을 처리할 때 그 순서를 문장으로 표현하면 문장이 길어지고 복잡해져서 이해하기 어렵지만 이를 도식하면 간결하고 명확하게 표현할 수 있으며 이해하기 쉬움
- 컴퓨터의 작업 대상이 되는 업무는 일반적으로 여러 요소가 포함되어 대단히 복잡한 것이 많으며 논리적으로 여러 갈래로 분기되는 것이 많음
- 순서도란 처리하고자 하는 문제를 분석하여 그 처리 순서를 단계화시켜 상호간의 관계를 일정한 기호를 사용하여 일목요연하게 나타낸 그림
- 순서도 작성의 역할
 - 문제 처리의 과정을 논리적으로 파악할 수 있어 정확성 여부를 쉽게 판단 가능
 - 문제를 해석하고 분석하며, 타인에게 전달 용이
 - 프로그램의 보관, 보수, 유지의 자료로서 활용이 가능
 - 프로그램 코딩의 기본 자료
- 컴퓨터 처리에 있어서 자료 처리과정을 나타낼 수 있는 순서도는 크게 일반 순서도(general flowchart)와 상세 순서도(detailed flowchart)로 구분



1.1 순서도의 개요_8

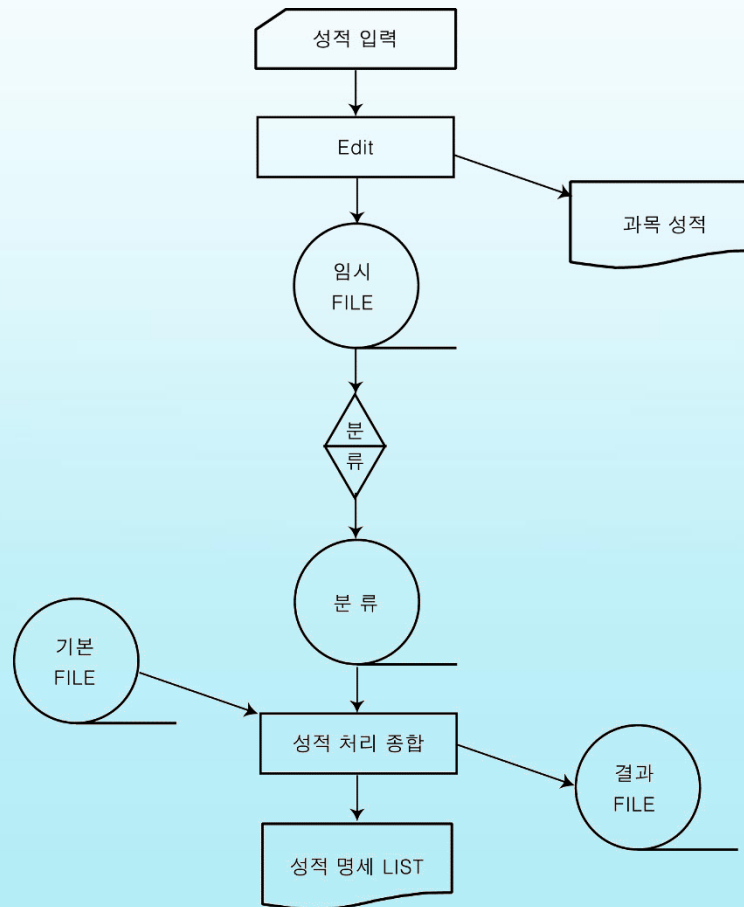
● 순서도의 종류

■ 시스템 순서도

- 컴퓨터에 의하여 어떤 일을 하고 있는가를 도식화한 것
- 원시자료가 최종정보로 변환되는 과정을 컴퓨터의 처리과정을 중심으로 알기 쉽게 표시한 것
- 시스템 순서도를 보면 어떤 자료가 사용되고 있는가, 주변장치가 어떤처리과정에서 어떻게 사용하고 있는가, 어떤 보고서가 작성되는가 등을 한 번에 판단 가능
- 시스템 순서도는 시스템 설계의 초기단계에서는 작업 내용을 총괄적으로 검토하기 위하여 사용
- 최종단계에서는 프로그래밍 작업과 연결시켜 주는 중요한 역할 수행
- 시스템 순서도의 작성 목적
 - 대상 업무에 관한 처리 과정의 전체적인 연관성을 파악
 - 작성한 프로그램의 수를 파악
 - 작성된 프로그램의 작업 순위 파악
 - 대상 업무의 전체적인 실행번호, 입력번호, 출력번호 정함
 - 기계처리 시간을 산출

1.1 순서도의 개요_9

- 시스템 순서도의 작성법
 - 컴퓨터를 활용하여 업무를 처리할 때 업무 전체의 흐름을 파악하기 위하여 시스템 순서도를 작성해야 한다
 - 시스템 순서도는 업무의 각 단위별로 크게 나누어 표시
 - 성적입력과 계산 총합 분류 및 명세 작업을 하는 과정





1.1 순서도의 개요_10

- 일반 순서도(general flowchart 또는 preliminary design)
 - 시스템 순서도가 작성되면 프로그래밍 하기 위하여 프로그램 순서도를 작성
 - 시스템 순서도가 <어떤 일을 하는가>를 표시하는 것이라면, 프로그램 순서도는 <일을 어떻게 할 것인가>를 표시한 것
 - 프로그램 순서도는 시스템 순서도에 표시된 하나의 전산처리 단위별로 작성
 - 프로그램 순서도는 그것이 어느 정도 세밀한가에 따라서 일반 순서도(general flowchart)와 상세 순서도(detailed flowchart)로 구분
 - 일반 순서도는 하나의 전산처리 단위의 내용을 개괄적으로 도식하여 프로그램 진행순서를 표시하는 것
 - 논리적인 잘못을 발견하기 쉽게 즉, 프로그램 전체의 내용 설명을 표현하기 위한 것
 - 일반 순서도가 작성되어 논리적인 모순이 없다고 판단되면 이것을 기준으로 세부 처리순서를 표시한 상세 순서도를 작성



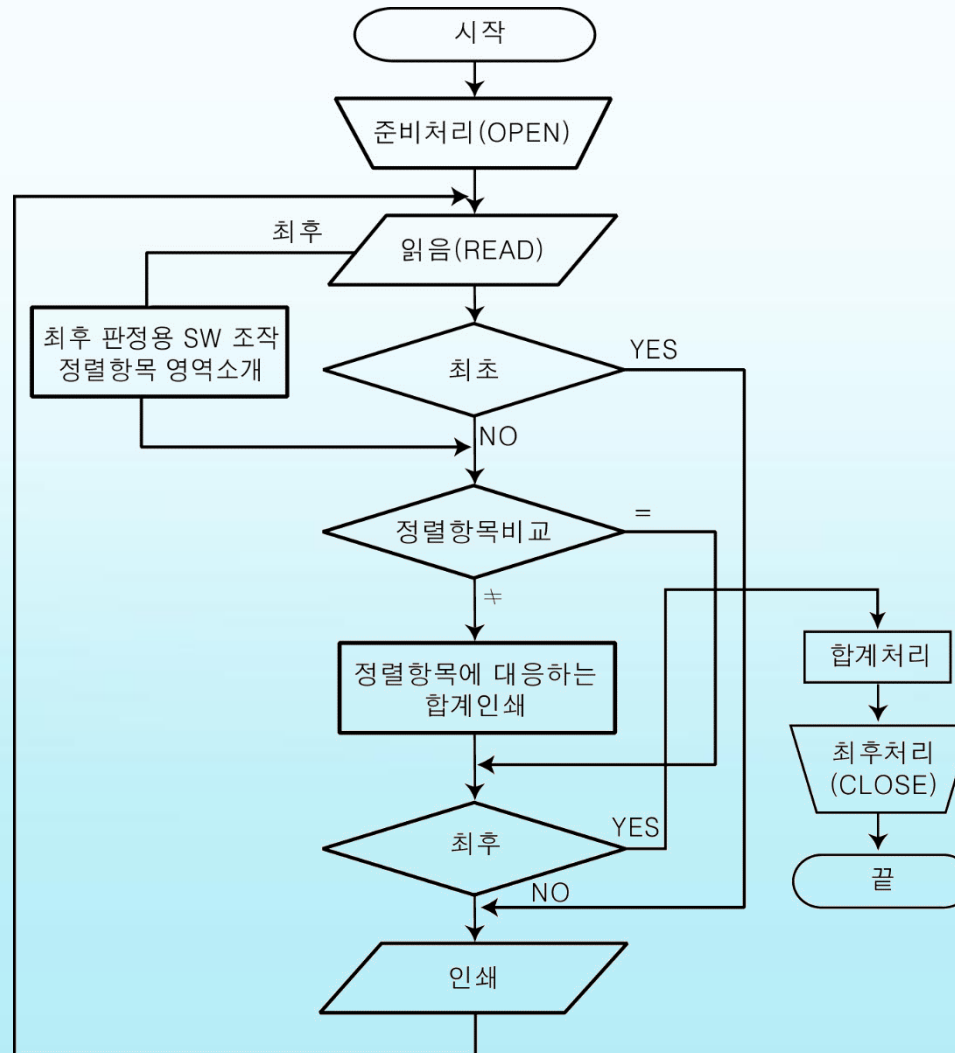
1.1 순서도의 개요_11

- 상세 순서도(detailed flowchart)
 - 코딩을 하기 위하여 최종적으로 작성된 순서도
 - 용도
 - 프로그램 작성 상 최종 검토 자료
 - 코딩 자료
 - 프로그램의 논리적 모순 검토 자료
 - 문서 자료(documentation)
 - 일반 순서도 : 프로그램 전체의 내용을 개괄적으로 설명한 것으로 시스템의 설계단계에서 검토 작성이 됨
 - 상세 순서도 : 개요 순서도를 좀 더 세분화한 것으로 프로그램 설계 단계에서 작성되며, 프로그램 코딩을 할 수 있는 형태로 되어 있음
 - 상세 순서도 작성 방법
 - 순서도에 의한 프로그램 진행 순서가 확인되고 논리적으로 이상이 없다고 인정되면 프로그래밍에 사용될 상세 순서도를 작성
 - 상세 순서도는 코딩을 위하여 작성하는 것이므로 프로그래머가 입출력 설계서를 보고 내용을 바로 이해하여 프로그래밍할 수 있도록 상세히 작성



1.1 순서도의 개요_12

- 성적입력과 계산 총합 분류 및 명세 상세 순서도



1.2 순서도의 기본형_1

- 순서도의 기본형에는 직선형, 선택형, 반복형 구조를 생각해 볼 수가 있음
- 직선형(sequence) 구조란 순차적으로 수행되어지거나 실행되어지는 연속적인 단계를 의미
- 두 변수들의 값을 교환시킨다는 것의 알고리즘은 순서 구조 이용
- A와 B의 변수일 때 알고리즘
 - A와 B의 값을 얻음
 - A의 값을 세 번째 변수 TEMP에 할당
 - B의 값을 A에 할당
 - TEMP의 값을 B에 할당
 - A와 B의 값을 나타냄

```
1  main() {  
2      int a, b, temp;  
3      scanf("%d %d", &a, &b,);  
4      temp = a;  
5      a = b;  
6      b = temp;  
7      printf("%d %d", a, b);  
8  }
```

1.2 순서도의 기본형_2

- 알고리즘은 두 변수의 값을 교환하고자 하는 경우 언제든지 수행
- 알고리즘 상에서 더 많은 유연성이 필요하게 될 것이며 선택(selection) 구조는 이 같은 사양을 처리
- 자동 예금인출기의 알고리즘
 - 거래(고객으로부터)의 종류와 금액(양수로 AMT)을 얻음
 - 고객의 현재 잔고(BAL, 은행 기록상으로부터)를 얻음
 - 만약 거래가 예금이면, BAL에 AMT를 더하고, 출금 경우는 BAL에서 AMT만큼 뺀
 - 거래금액 AMT과 새롭게 변경된 잔고 BAL을 표시

```
1  main() {  
2      int sel, amt, balance;  
3      scanf("%d %d ", &sel, &amt);  
4      read balance; // 은행 컴퓨터로 부터 읽어 들임  
5      if (sel == 1)  
6          balance = balance - amt;  
7      else  
8          balance = balance + amt;  
9      write balance; // 은행 컴퓨터에 저장  
10     printf("%d %d", amt, balance);  
11 }
```

1.2 순서도의 기본형_3

- 알고리즘은 금액 AMT가 양수로서 되어있다고 설명되어져 있으나 음수가 발생되리라고 생각해서 그러한 오류를 방지키 위한 프로그램 작성 필요
- 반복수행(iteration)은 여러 번 실행되어야 할 일련 단계들을 이용
- 루프(loop)란 말은 가끔 되풀이되는 단계를 기술하기 위해서 사용
- 반복수행 구조의 공통적인 응용은 얼마나 많이 처리되어지는가 하는 상태를 파악하는 것
- 학생들의 학점평균을 계산하는 프로그램은 각 학기당 다른 수많은 학생의 성적들을 처리
- 컴퓨터는 등급표를 계산하게 되며 계산을 위해 한 변수가 필요한데, 여기서 그것을 카운터(counter)라 부르고 그리고 제일 처음에는 그 값을 0으로 할당
- 이전에 그 변수에 어떤 값이 있든지 관계치 않으며 한 변수가 어떤 계산에서 사용되기 전에 한 값을 할당시키는 것을 변수로 초기화(initialize)시킨다고 함

1.2 순서도의 기본형_4

▪ 알고리즘

- COUNTER라고 명명한 변수에 0의 값을 할당시킨다.
- 마지막 학생성적이 처리될 때까지 단계 3과 4를 반복시킨다.
- 학생성적을 받아서 처리한다.
- COUNTER에 1을 더한다.
- COUNTER의 값을 표시한다.

- 반복수행이 사용되어질 때마다, 다시 루핑(looping)을 할 것인지 아니면 그 루프를 벗어날 것인지 하는 판단이 이루어지게 될 것
- 그러한 시험을 조건(condition)이라하며 실행에 대한 조건이 그 점에서 만나게 되는 한, 루핑이 실행
- 첫째 조건은 루프에 들어가기 전에 초기화되어져야만 함
- 둘째 조건은 루프 내에서 변경이 되어져서 결국에는 루프가 끝
- 컴퓨터 프로그래밍 하는 과정에서 가끔은 연속적인 수를 더하는 것을 필요로 함
- 처리되는 전체 결과는 누적변수라 불리는 변수에 누적

1.2 순서도의 기본형_5

- 첫 단계는 이전 문제에서와 같이 누적변수를 초기화시키는 것
- 실행에 대한 조건은 상당히 많은 수들이 존재하는 것
- 컴퓨터는 마지막 항목이라는 것을 인식할 수가 있음
- dummy값이 입력의 끝을 나타내는데 이용되며 end-marker라 부름
- 알고리즘
 - TOTAL에 0의 값을 할당시킴(TOTAL을 초기화)
 - 첫 수 NUM을 얻음
 - NUM이 end-marker가 아닌 경우 4번과 5번 단계를 수행
 - NUM을 TOTAL에 더함
 - 다음 수를 얻어서 그것을 NUM에 할당
 - TOTAL의 값을 표시

```
1  main() {  
2      int num, total = 0;  
3      scanf("%d", &num);  
4      while(num != EOF) {  
5          total = total + num;  
6          scanf("%d", num);  
7      }  
8      printf("%d", total);  
9 }
```



1.2 순서도의 기본형_6

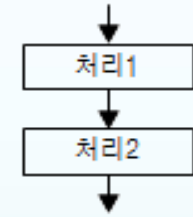
- end-marker가 단계 5에서 만나게 된 후, 실행에 대한 조건이 루프 (단계 3)의 시작에서 비교
- 조건이 맞지 않는 경우는 루프를 벗어나고, 실행은 단계 6을 계속
- TOTAL의 값은 단계 6이 루프 밖에서 발생되기 때문에 단지 나타남
- 알고리즘을 다음과 같은 방법으로 작성한다면, TOTAL의 값은 한 수가 더해질 때마다 표시
 - TOTAL에 0의 값을 할당시킴(즉, 초기화)
 - 첫 수 NUM을 얻음
 - NUM이 end-marker가 아닌 경우, 단계 4에서 6까지 수행
 - NUM을 TOTAL에 더함
 - TOTAL의 값을 나타냄
 - 다음 수를 받아들여, 그 값을 NUM에 할당

1.2 순서도의 기본형_7

● 일반적인 알고리즘의 종류

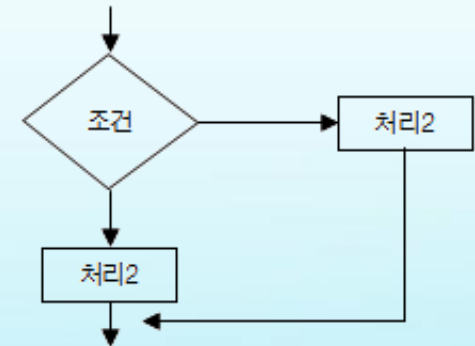
■ 직선형

좌우로 갈라지는 일이 없이 아래로만 내려가며 진행되는 순서대로 표현



■ 선택형

- 조건을 판단할 때나 비교할 때 의사결정 기호를 사용
- 기호 내에 기준을 쓰며 2가지 조건으로 갈라질 경우에는 오른쪽으로 분기시켜서 흐름선이 겹치지 않고 단순하게 됨



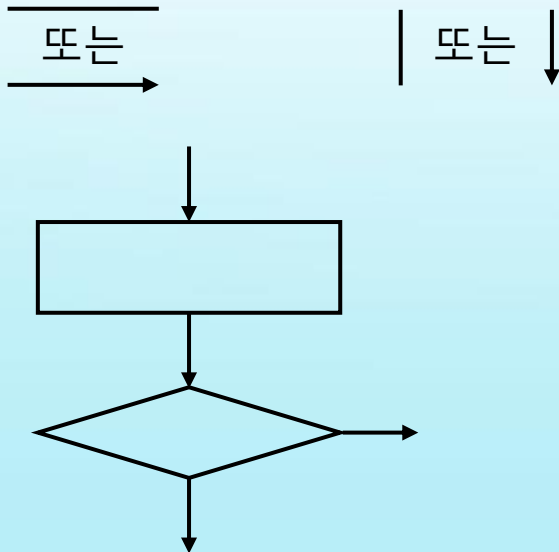
■ 반복형

어떠한 조건이 만족될 때까지 일정부분을 되풀이하고 설정된 조건이 맞으면 아래로 흐름



1.3 순서도에 이용되는 기호_1

- 순서도에 사용되는 기호나 그 의미는 통일되어 있지 않으면 목적 달성을 할 수가 없게 됨
- ISO(국제표준화기구 : International Standard Organization)의 권고안(97N90)에 의해 나열된 기호
- 흐름선 기호
작업의 흐름 방향 표시



1.3 순서도에 이용되는 기호_2

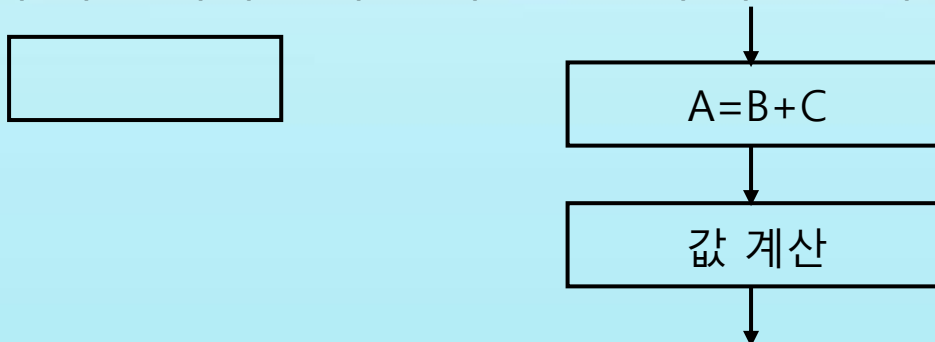
▪ 단말 기호

순서도의 시작과 정지를 나타내는 것으로 항상 이 기호에서 시작되어서 끝이 남



▪ 처리 기호

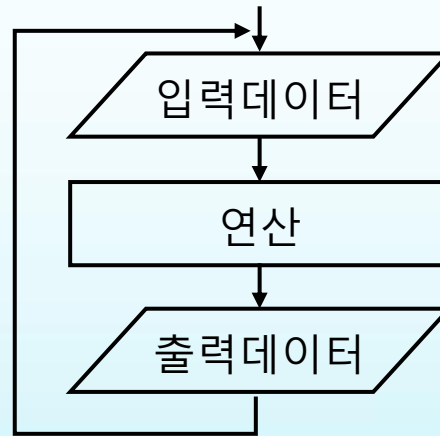
- 여러 가지 종류의 작업을 처리하는 것을 뜻함
- 이 기호 내에는 작업 내용을 표시 하는 문자들을 기입



1.3 순서도에 이용되는 기호_3

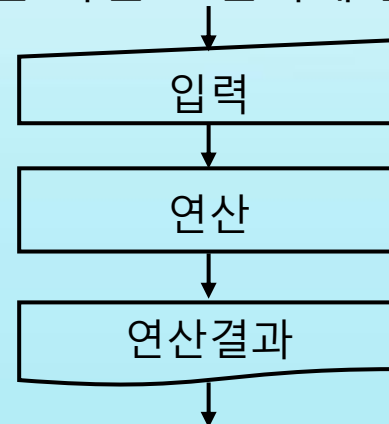
■ 입출력 기호

- 모든 종류의 입출력 작업을 나타내는 것을 의미
- 모든 종류의 입출력의 매개체가 불명확하거나 또는 매체를 명시할 필요가 없을 때 이용



■ 프린터 기호

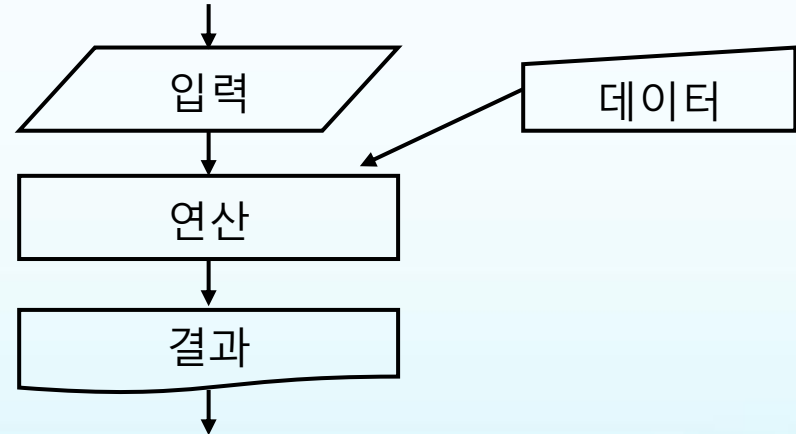
출력을 나타내고 기호로서, 문서를 라인 프린터에 출력시키는 것



1.3 순서도에 이용되는 기호_4

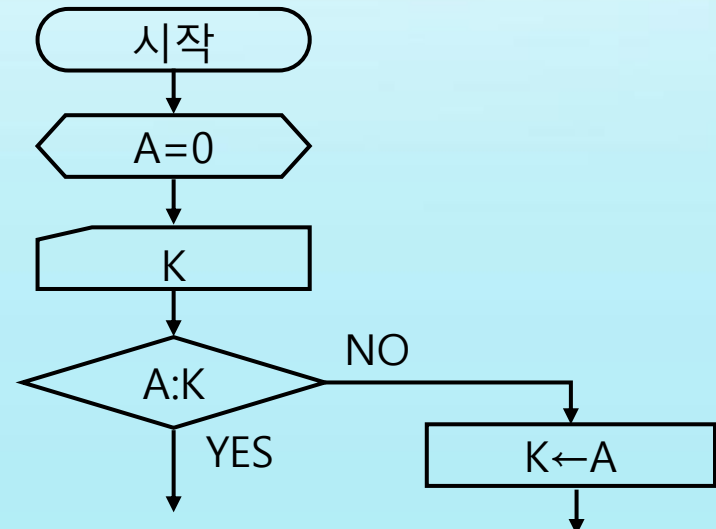
단말 기호

터미널에 의한 입/출력 기능을 나타내는 것



준비 기호

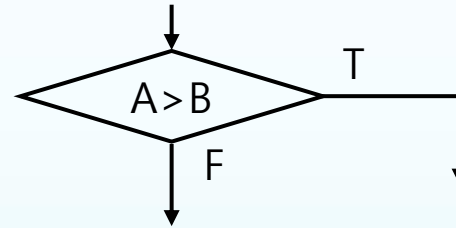
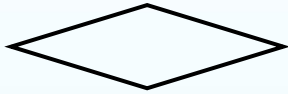
처리에 앞서서 처리에 필요한 조건을 결정하는 기호



1.3 순서도에 이용되는 기호_5

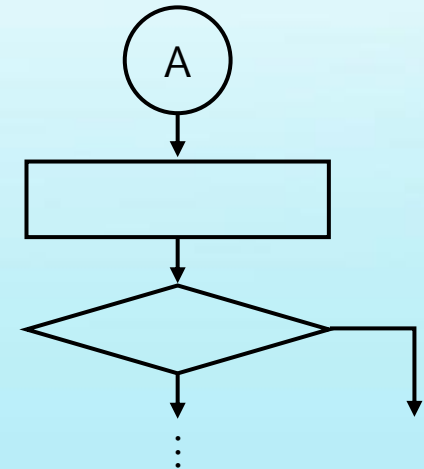
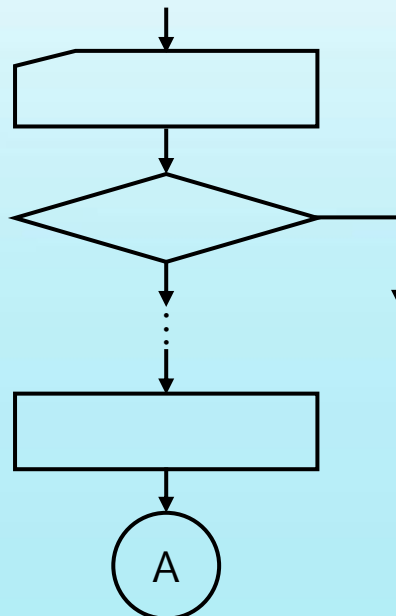
■ 판단 기호

여러 가지 경로 중에 어느 경로를 택할 것인가를 결정하는 기호



■ 결합 기호

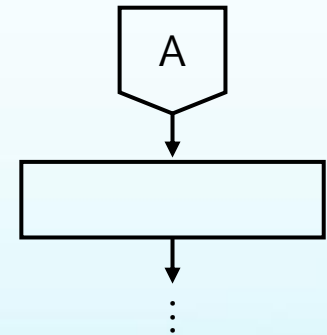
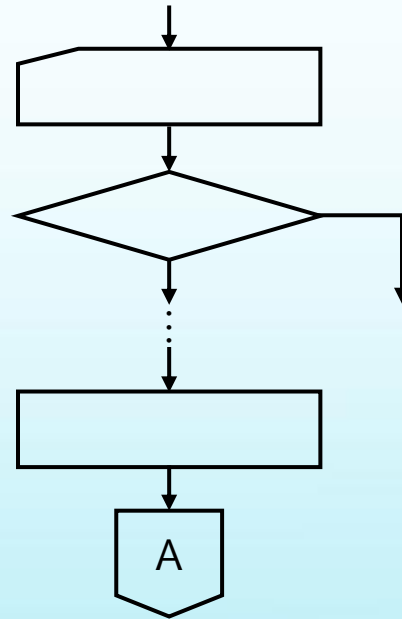
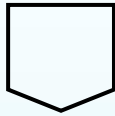
동일한 페이지 내에서 순서도나 다른 부분으로 옮겨질 때 이 기호 내에 숫자나 문자를 이용해서 상호 연결해주는 기호



1.3 순서도에 이용되는 기호_6

▪ 페이지 연결 기호

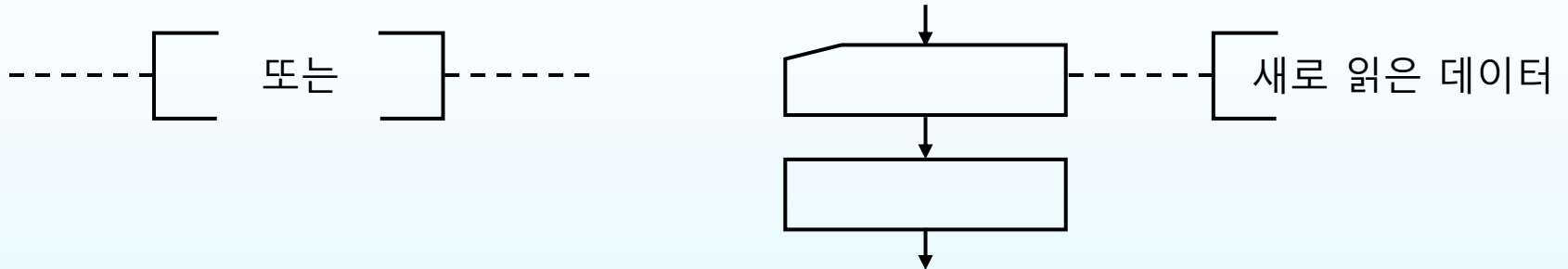
순서도 전체를 한 페이지 내에 모두를 그릴 수 없는 경우에 다른 페이지에 계속해서 그릴 때 이 기호 내에 숫자나 문자를 이용하여 연결



1.3 순서도에 이용되는 기호_7

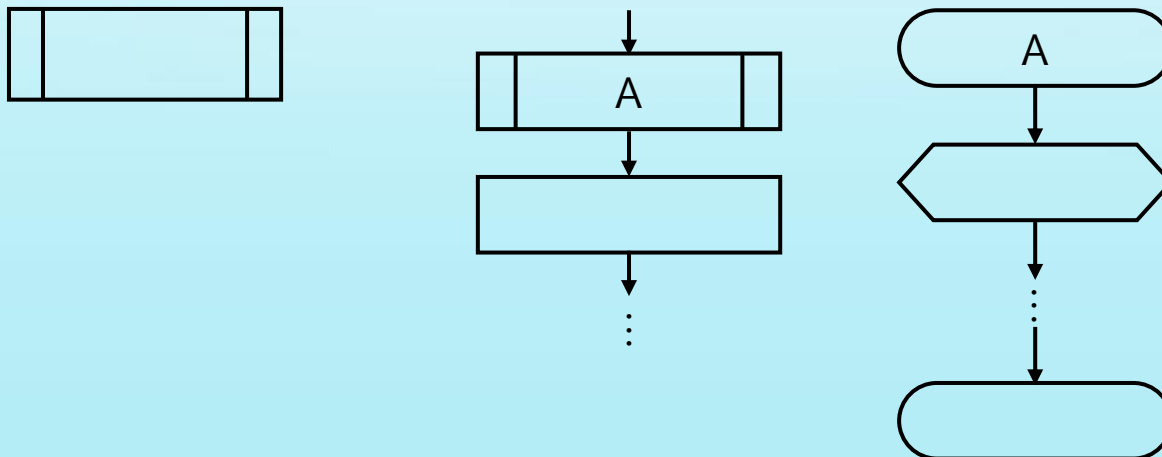
■ 주석(annotation, 또는 comment) 기호

작업 내용 또는 처리 하고자 하는 내용에 대한 설명을 부가적으로 하는 경우에 이용



■ 사전에 정의된 처리 기호

- 반복되어지는 내용이 있는 경우에 다른 일에 반복되어지는 순서도를 그리고 그것을 부르기 위해서 이용
- 주로 서브루틴처럼 별도의 처리 프로그램에 이용


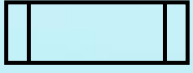


1.3 순서도에 이용되는 기호_8

▪ 순서도 기호 정리




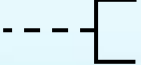

기 호	이 름	의 미
	터미널	순서도의 시작과 끝을 표시
	준비	배열선언 및 초기 설정에 사용
	흐름선(Flow-line)	순서도 기호간 연결 및 흐름을 표시
	반복(Loop)	반복 수행
	자기디스크	자기디스크를 매체로 사용
	종이테이프	종이테이프를 정렬할 때 사용
	분류(Sort)	데이터를 정렬할 때 사용
	표시(Display)	화면으로 출력
	출력(Print)	프린트 등으로 출력

1.3 순서도에 이용되는 기호_9

기 호	이 름	의 미
	수작업	Off-line 작업
	통신	통신회선으로 접속
	온라인(On-line)기억	온라인 보조기억장치
	입출력	데이터의 입출력시 사용
	비교, 판단	비교 및 판단에 의한 논리적 분기를 할 경우 사용
	결합	같은 페이지에서 순서도 흐름을 연결
	서브루틴	부 프로그램 처리
	자기테이프	자기테이프 매체로 사용
	펀치 카드	카드를 매체로 사용



1.3 순서도에 이용되는 기호_10

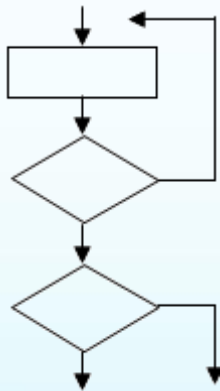
기 호	이 름	의 미
	조합(Merge)	여러 개의 파일을 하나로 합침
	수작업 입력	콘솔(Console)에 의한 입력
	페이지 결합	순서도 흐름이 다른 페이지로 연결될 경우 사용
	주석	주석이나 설명을 표시
	오프라인(Off-line)기억	오프라인 기억장치



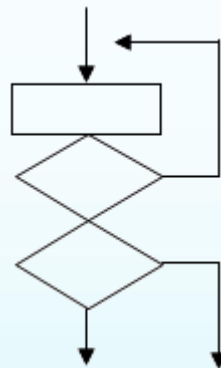
1.4 순서도 작성 요령_1

■ 기호 간의 연결

기호 간은 반드시 흐름선으로 연결



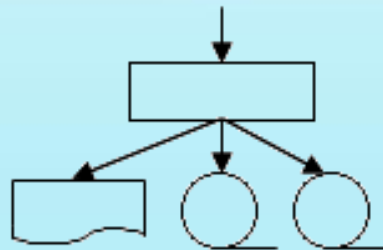
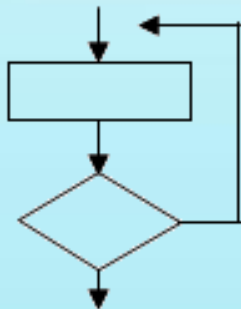
바름



틀림

■ 흐름 방향

- 흐름 방향은 원칙적으로 상(上)에서 하(下)로, 좌(左)에서 우(右) 표시
- 사선 또는 루프(loop)를 형성할 때는 예외



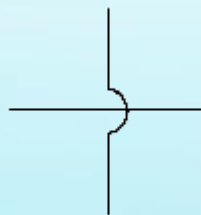
1.4 순서도 작성 요령_2

■ 화살 표시

- 흐름 방향을 표시하는 화살표는 원칙적으로 모든 흐름선 선단에 표시
- 원칙과 일치할 때는 화살표를 생략할 수도 있음
- 흐름선 선단 부근의 기록 밀도가 클 경우 또는 흐름선이 굴절되어 길게 연결될 때는 흐름선 중간에 화살표를 표기할 수도 있음

■ 흐름선의 교차

- 흐름선은 가능한 한 상호 교차되지 않도록 그려야 함
- 불가피한 경우에는 다음과 같이 작성



바름



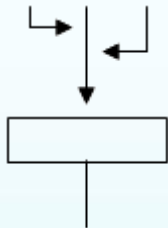
틀림



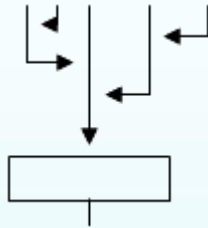
1.4 순서도 작성 요령_3

■ 흐름선의 결합

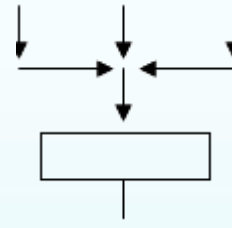
- 두 개 이상의 흐름선이 결합할 때는 아래/그림과 같이 표시
- (c)와 경우는 흐름선의 교차와 구분하기 위하여 결합점에 화살 표시



(a)



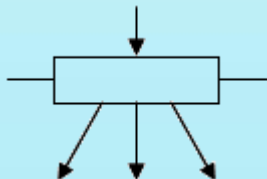
(b)



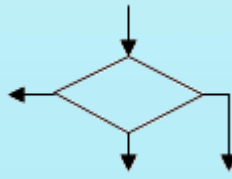
(c)

■ 흐름선의 분기

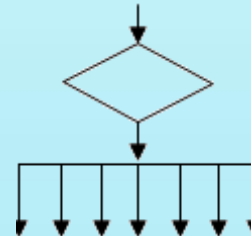
- 하나의 기호로부터 여러 개의 분기선이 나갈 경우에는 다음 그림과 같이 표시
- 분기선이 많을 경우에는 (c)와 같이 표시하며 이 때 흐름선의 교차와 구분하기 위하여 교차점에서 화살 표시



(a)



(b)



(c)



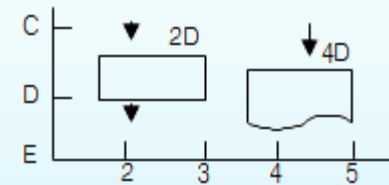
1.4 순서도 작성 요령_4

■ 기호 내에 정보의 기입

기호내의 정보를 기입할 때는 정보의 흐름과는 관계없이 위에서 아래로
좌에서 우로 기입

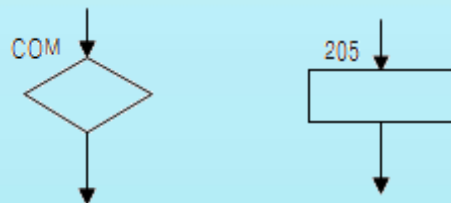
■ 기호 위치의 표시

- 기호 위치의 좌표를 표시할 때는 해당 기호의
우상단에 표시
- 숫자 좌표를 문자 좌표보다 먼저 기록
- 좌표는 기호의 동거리일 때는 좌상(左上)의
좌표를 사용



■ 표식 표시

흐름도와 프로그램의 관련을 표시하기 위한 표식(레이블(label), 패러그
래프명(paragraph name), 문번호(statement number))은 기호의 좌상단
에 표시



1.5 순서도 프로그램

- 순서도를 그리는 도구
 - <http://dia-installer.de/>

