

4주차

- 알고리즘 소개
- 알고리즘의 표현 방법
- 의사코드
- 흐름도

		강의 주제	주 차
1부 컴퓨팅 사고력	컴퓨팅 사고력의 소개	IT 사회, 소프트웨어 세상, 컴퓨팅 사고력의 소개, 컴퓨팅 사고력의 개념	1
		컴퓨팅 사고력의 개념, 주위에서 볼 수 있는 컴퓨팅 사고력, 문제해결 방법	2
	문제해결 방법, 컴퓨터	문제해결 방법, 문제해결 과정 예, 문제해결을 위한 소프트웨어 설계 사상, 컴퓨터의 특징, 소프트웨어, 유한상태기계	3
2부 소프트웨어	알고리즘	알고리즘 소개, 알고리즘의 표현 방법, 의사코드, 흐름도	4
	프로그램	프로그램의 기능, 함수, 컴파일러	5
	파이썬	파이썬 소개 및 설치, 변수에 값 저장, 입력, 출력, 조건부 수행	6
		반복, 리스트, 함수, 출력 형식	7
3부 컴퓨팅 사고력 활용하기	데이터의 표현	이진수, 아스키코드, 오디오 데이터, 이미지 데이터, 자료구조	8
		인코딩 및 압축, 오류확인	9
	데이터의 저장과 검색	배열 및 연결 리스트, 선형검색, 이분검색, 색인순차검색, 해싱	10
		이진검색트리, 최대값 및 최소값 검색	11
	알고리즘설계	정렬, 분할정복 알고리즘, 탐욕적 알고리즘	12

알고리즘 단어의 기원

al-Khwarizmi의 1200번째 생일을 기념하기
위해 구 소련에서 1983년에 발행한 우표



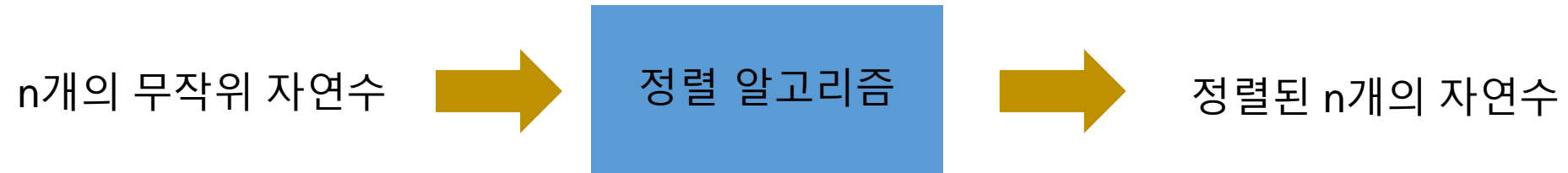
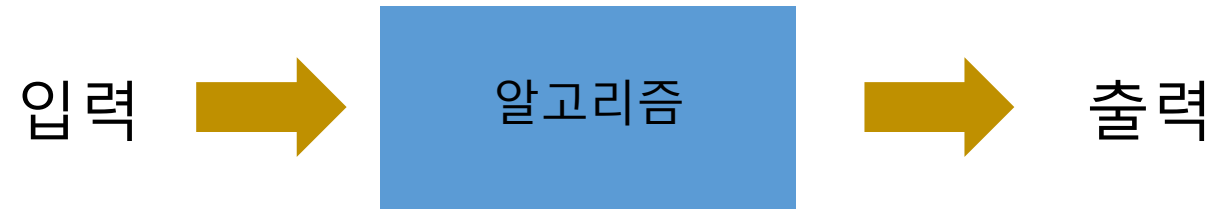
- al-Khwarizmi (780-850)
- 페르시아의 수학자, 천문학자, 지리학자

정의: 알고리즘(algorithm)

- 문제를 해결할 수 있는 잘 정의된(well defined) 유한 (finite)시간 내에 종료되는 계산적인(computational) 절차(procedure)
 - 1) 잘 정의된: 누구든지 정확하게 이해할 수 있을 정도로 기술되어야 한다.
 - 2) 유한시간 내에 종료: 무한시간의 반대 개념
 - 3) 계산적인 절차: 계산적으로 수행될 수 있어야 한다.



- 입력을 받아서 출력으로 전환시켜주는 일련의 계산절차



프로그램



- 알고리즘은 프로그램의 엔진에 해당하는 중요한 절차이다.



Q1. 24과 16에 대해

- 최대공약수?
- 최소공배수?

- 우리는 각 문제를 해결하는 절차를 알고 있다.

2	24	16
2	12	8
2	6	4
	3	2

$$\text{최대공약수: } 2 \times 2 \times 2 = 8$$

$$\text{최소공배수: } 2 \times 2 \times 2 \times 3 \times 2 = 48$$

- 알고 있는 이 방법을 어떻게 설명할 수 있나?



- 소인수분해 방법

$$24=2 \times 2 \times 2 \times 3=2^3 \times 3^1, 16=2 \times 2 \times 2 \times 2=2^4$$

$$\text{최대공약수} = 2^{\min\{3,4\}} \times 3^{\min\{1,0\}} = 2^3 \times 3^0 = 8$$

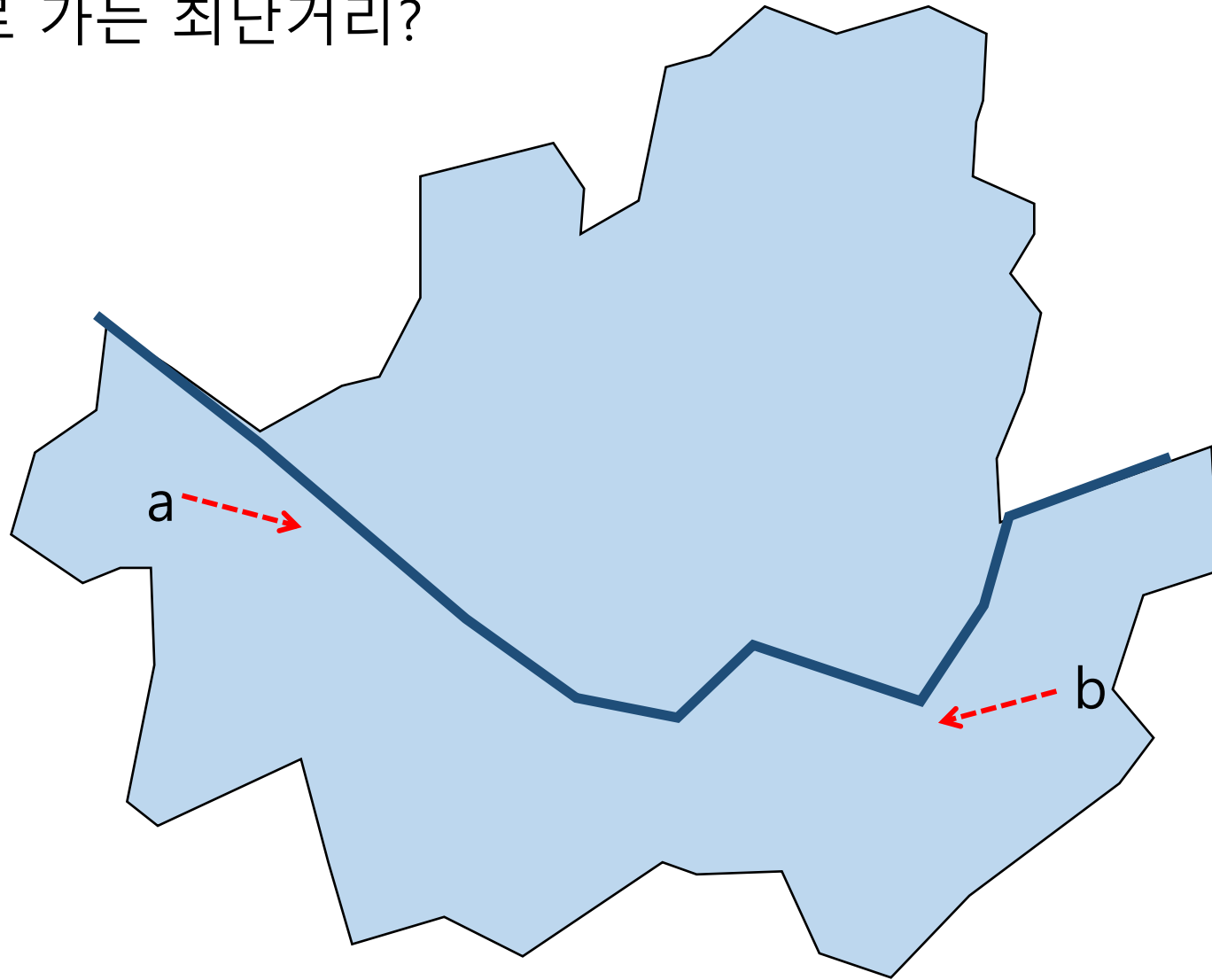
$$\text{최소공배수} = 2^{\max\{3,4\}} \times 3^{\max\{1,0\}} = 2^4 \times 3^1 = 48$$

- 유클리드 알고리즘

✓ 두 수 a, b 의 최대공약수를 $\text{GCD}(a,b)$ 라 하면, $\text{GCD}(a,b) = \text{GCD}(a-b,b)$,
 $a > b$ 인 경우, $\text{GCD}(a,b) = \text{GCD}(a,b-a)$, $a < b$ 인 경우, $\text{GCD}(a,a)=a$ 성질을
이용하여 두 수의 최대공약수를 구할 수 있다.

✓ 예를 들어, $\text{GCD}(24,16) = \text{GCD}(8,16) = \text{GCD}(8,8)$ 이 되어, 8을
최대공약수로 찾을 수 있다.

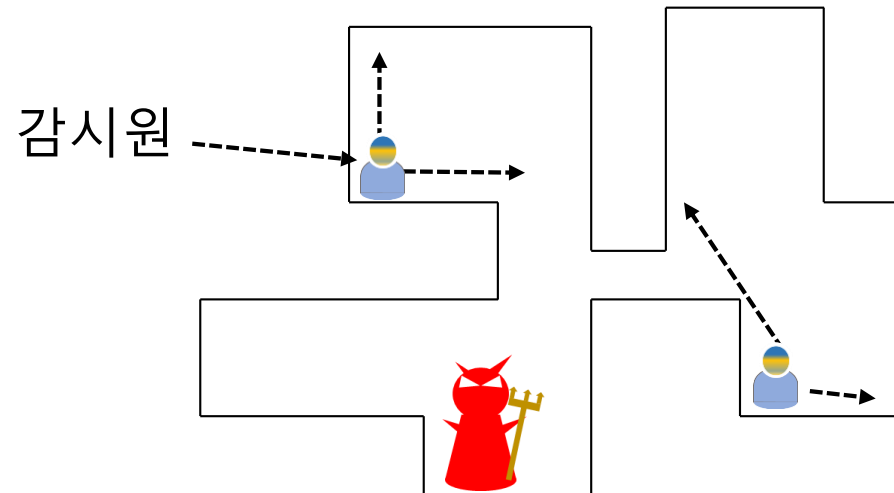
Q2. a에서 b로 가는 최단거리?



다익스트라(Dijkstra) 알고리즘 사용

Q3. 화랑(Art Gallery)감시 문제

모든 공간을 감시하기 위한 최소 인원 배치는?

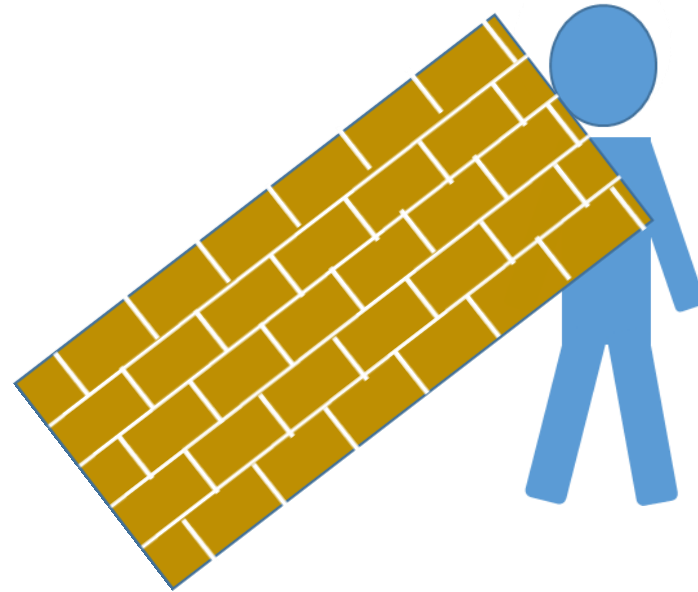
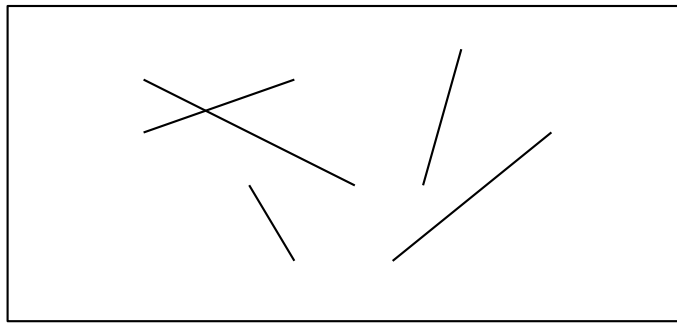


화랑공간



Q4. 선분교차 문제

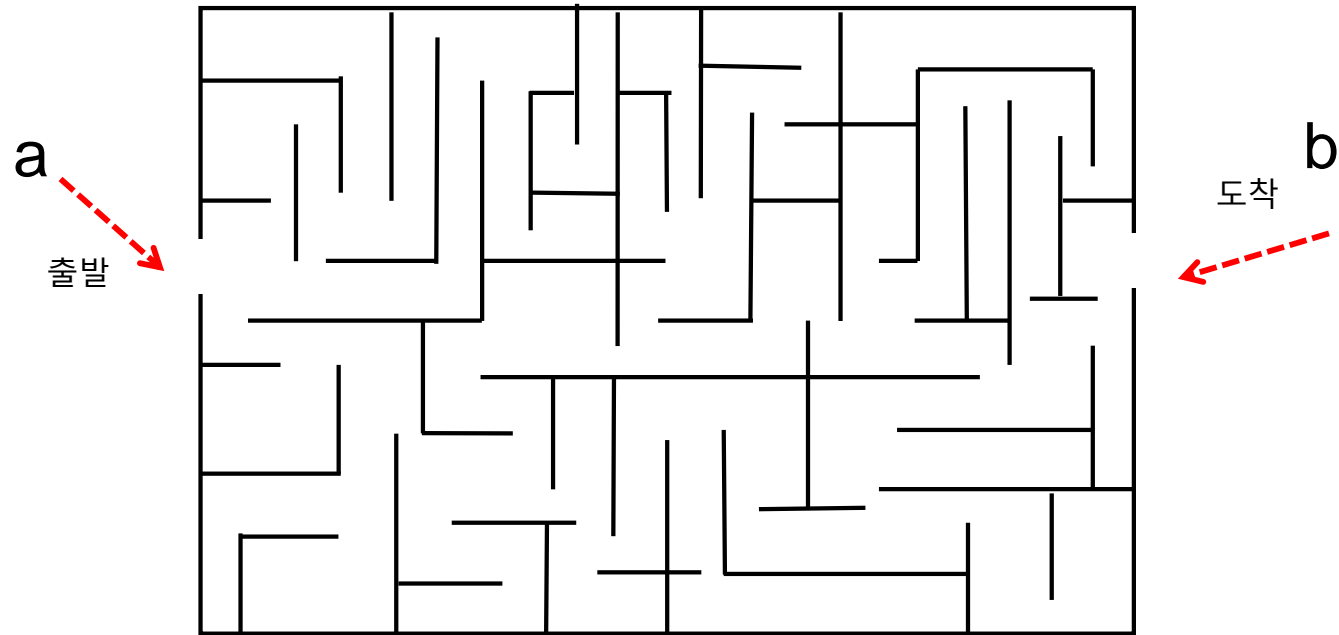
- 컴퓨터 그래픽에서 평면 또는 3차원 공간상의 두 선분이 교차하는지 확인하는 방법은?
- 평면과 선분이 어느 지점에서 교차하는지 확인하는 방법은?



벽과 사람의 출동 장면 표현



Q5. a에서 b로 가는 길?



Q6. 데이터를 크기가 증가하는 순서로 정리한다.

- 정렬(sort) 알고리즘

학교	강	자동차	국가	운전	전화기	컴퓨터	가방	책
----	---	-----	----	----	-----	-----	----	---

정렬



가방	강	국가	운전	자동차	전화기	책	컴퓨터	학교
----	---	----	----	-----	-----	---	-----	----



알고리즘의 표현 방법

1. 자연어(natural language):

- ✓ 사람이 사용하는 언어를 이용
- ✓ 단어의 이해가 서로 다를 경우 의미가 정확히 전달될 수 없다.
- ✓ 절차가 긴 알고리즘인 경우 많은 문장이 필요하다.
- ✓ 복잡한 알고리즘 표현방법으로는 효과적이지 않다.

2. 의사코드(pseudocode)

실제 코드는 아니지만, 프로그램 코드와 비슷한 것

- ✓ 자연어로 표현한 것 보다는 구조를 갖고 있다.
- ✓ 프로그램보다는 추상적인 방법이다.
- ✓ 보다 정확한 이해를 도와 준다.



3. 흐름도(flowchart)

- ✓ 도형을 이용하여 수행해야 할 작업을 표시
- ✓ 작업의 순서를 나타낸다.
- ✓ 전체 작업을 쉽게 이해할 수 있다.

4. 프로그래밍 언어

- ✓ 컴퓨터로 수행할 수 있는 프로그램으로 알고리즘을 표현



1. 자연어(natural language) 사용

[최대공약수 구하는 유클리드 알고리즘]

입력으로 두 자연수 a, b 를 받는다. 만일 a 와 b 가 같으면 a 의 값을 출력한다. 만일 a 가 b 보다 크다면, $a-b$ 의 값을 새로운 a 의 값으로 설정한다. 만일 a 가 b 보다 크지 않으면, $b-a$ 의 값을 새로운 b 의 값으로 설정한다. 두 값이 같아 질 때 까지 이전 단계들을 반복한다. 같아진 값이 최대공약수이다.



[문제] 다음의 10개의 데이터에서 제일 큰 수를 찾는다.

0부터 인덱스 시작
C,C++언어 방식

a[0]	a[1]	a[2]	a[3]	a[4]	a[5]	a[6]	a[7]	a[8]	a[9]
5	3	8	2	9	4	7	1	0	6

- 배열(array)

공통의 성질을 갖는 변수가 여러 개 일 때 하나의 변수명을 정하고, 위치를 나타내는 인덱스를 이용해서 변수를 나타내는 자료구조 (data structure)



a[0]	a[1]	a[2]	a[3]	a[4]	a[5]	a[6]	a[7]	a[8]	a[9]
5	3	8	2	9	4	7	1	0	6

[배열에 저장된 10개의 수 중 최대값 찾기]

배열의 제일 처음에 있는 데이터를 임시로 최대값 M으로 설정한다. 오른쪽으로 이동하면서 보고 있는 데이터와 M과 비교해서, 보고 있는 데이터가 더 크면 그 데이터를 M으로 재설정하고, 아니면 다음의 오른쪽 데이터로 이동하면서 동일한 작업을 수행한다. 데이터의 끝까지 이동하면 M이 데이터의 최대값이 된다.



2. 의사코드(pseudocode)

[최대공약수 구하는 유클리드 알고리즘]

[알고리즘 최대공약수 찾기]

입력: 두 수 a, b

출력: 최대공약수

단계 0: 입력으로 두 자연수 a, b 를 받는다.

단계 1: 만일 $a=b$ 이면 a 의 값을 출력한다.

단계 2: 만일 $a > b$ 이면, $a=a-b$

그렇지 않다면, $b=b-a$

단계 3: 단계 1로 이동한다.

a[0] a[1] a[2] a[3] a[4] a[5] a[6] a[7] a[8] a[9]

5	3	8	2	9	4	7	1	0	6
---	---	---	---	---	---	---	---	---	---

[배열에 저장된 10개의 수 중 최대값 찾기]

[알고리즘 최대값 찾기]

입력: 배열 a[]에 저장된 10개의 데이터

출력: 최대값

(step 1) a[0]의 데이터를 임시로 최대값 M으로 설정. i=1

(step 2) a[i]의 값과 M과 비교해서, a[i]가 더 크면 a[i]를 M으로 재설정, 아니면 a[i+1] 값 비교로 이동

(step 3) step 2를 오른쪽 나머지 데이터에 대해서 수행. i = i + 1

(step 4) 데이터가 더 이상 없으면 M이 데이터의 최대값



- 또 다른 방식의 의사코드

- ✓ 의사코드를 작성하는 방법이 고정된 것은 아님
- ✓ 보다 프로그램에 가까운 의사코드

a[0]	a[1]	a[2]	a[3]	a[4]	a[5]	a[6]	a[7]	a[8]	a[9]
5	3	8	2	9	4	7	1	0	6

[배열에 저장된 10개의 수 중 최대값 찾기]

[알고리즘 최대값 찾기]

입력: 배열 a[]에 저장된 10개의 데이터

출력: 최대값

M = a[0]

for each a[i], $1 \leq i \leq 9$

 if a[i] > M then

 M = a[i]

 fi

end for

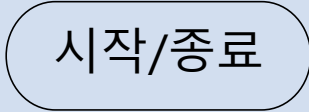
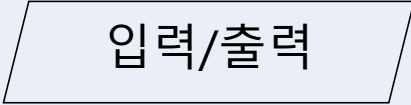
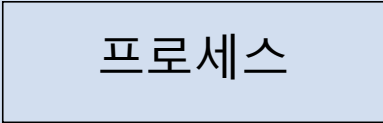
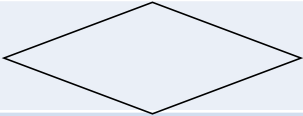
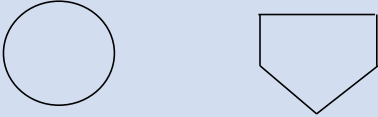


M is maximum

1. 들여쓰기
2. if – fi
3. for – end for

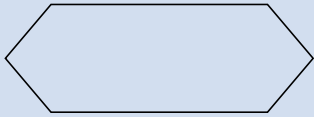
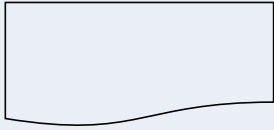



3. 흐름도(flowchart)

흐름도에서 사용되는 표시

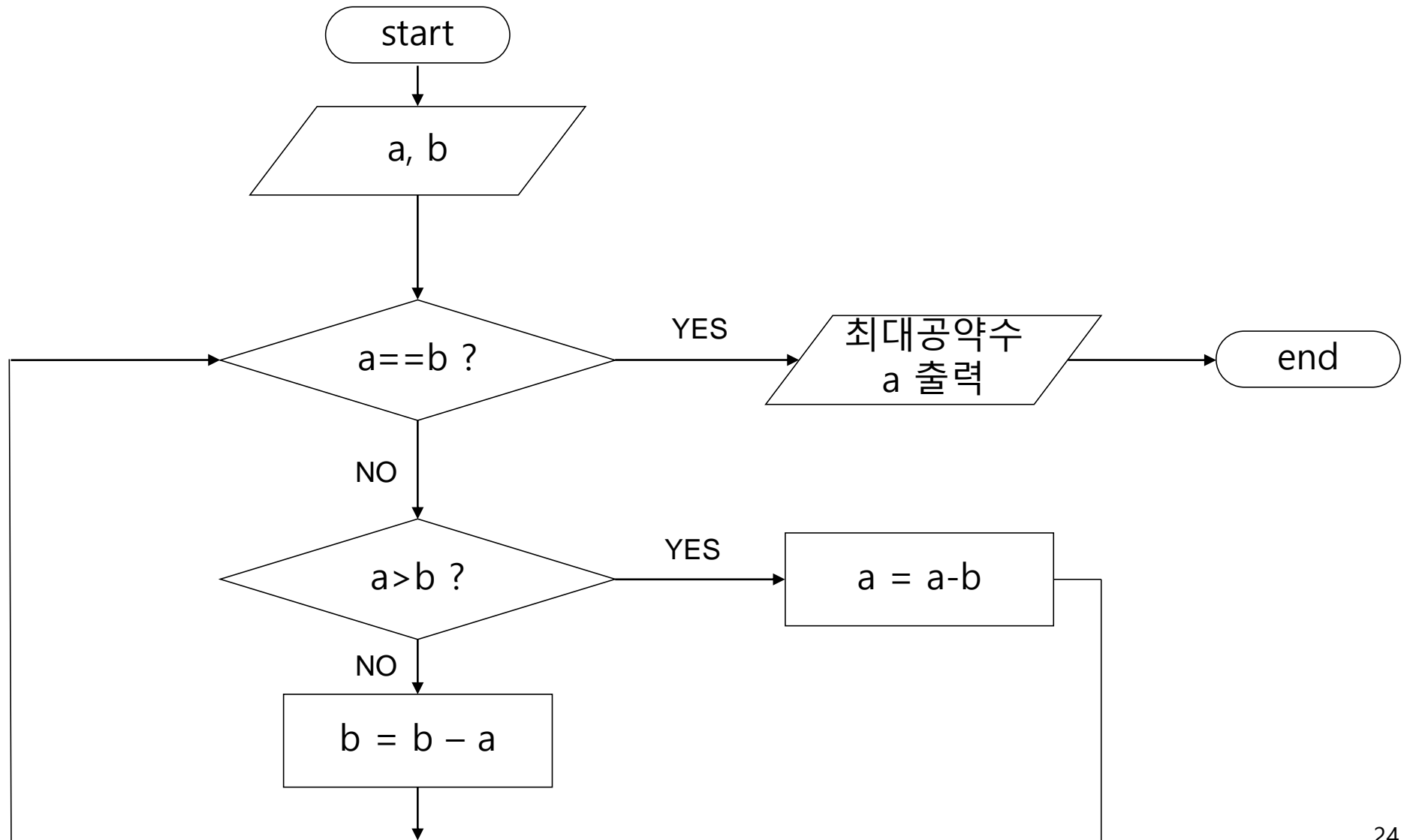
심벌 이름	도형	내용
터미널 심벌		흐름도의 시작과 끝을 나타내는 기호
입출력 심벌		입출력 작업 표시
프로세스 심벌		연산 명령문 등 처리해야 할 작업 내용
판단 심벌		판단을 나타낸다
연결심벌		페이지 내, 외의 연결을 나타낸다
미리 정의된 프로세스 심벌		모듈, 함수, 메소드, 하위절차
흐름심벌		연결 흐름 표시



심벌 이름	도형	내용
준비 심벌		변수의 초기화
문서출력 심벌		문서로 출력
데이터베이스 심벌		데이터베이스

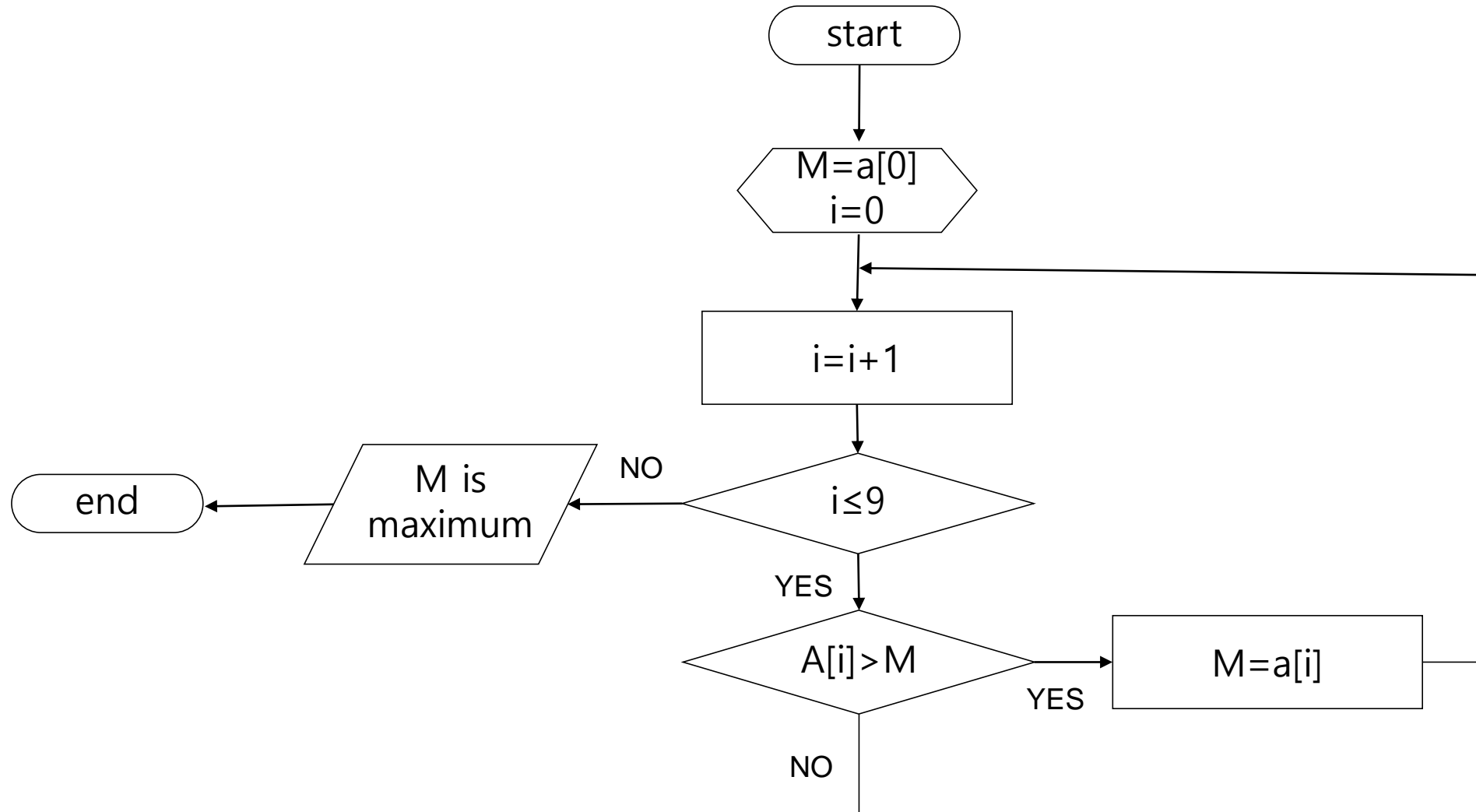


[최대공약수 구하는 알고리즘, 유클리드 방법]



[배열에 저장된 10개의 수 중 최대값 찾기]

a[0]	a[1]	a[2]	a[3]	a[4]	a[5]	a[6]	a[7]	a[8]	a[9]
5	3	8	2	9	4	7	1	0	6



4. 프로그래밍 언어(programming language)

[최대공약수 구하는 알고리즘, 유클리드 방법]

파이썬

```
a=int(input("A = "))
b=int(input("B = "))
while a!=b:
    if(a>b):
        a=a-b
    else:
        b=b-a
print("최대공약수 = ",a)
```



[배열에 저장된 10개의 수 중 최대값 찾기]

a[0]	a[1]	a[2]	a[3]	a[4]	a[5]	a[6]	a[7]	a[8]	a[9]
5	3	8	2	9	4	7	1	0	6

C++

```
void main() {  
    int i, a[10]={5,3,8,2,9,4,7,1,0,6};  
    int M;  
    M=a[0];  
    for(i=1;i<=9;i++)  
        if(a[i]>M) M=a[i];  
    cout << M;  
}
```



[배열에 저장된 10개의 수 중 최대값 찾기]

a[0]	a[1]	a[2]	a[3]	a[4]	a[5]	a[6]	a[7]	a[8]	a[9]
5	3	8	2	9	4	7	1	0	6

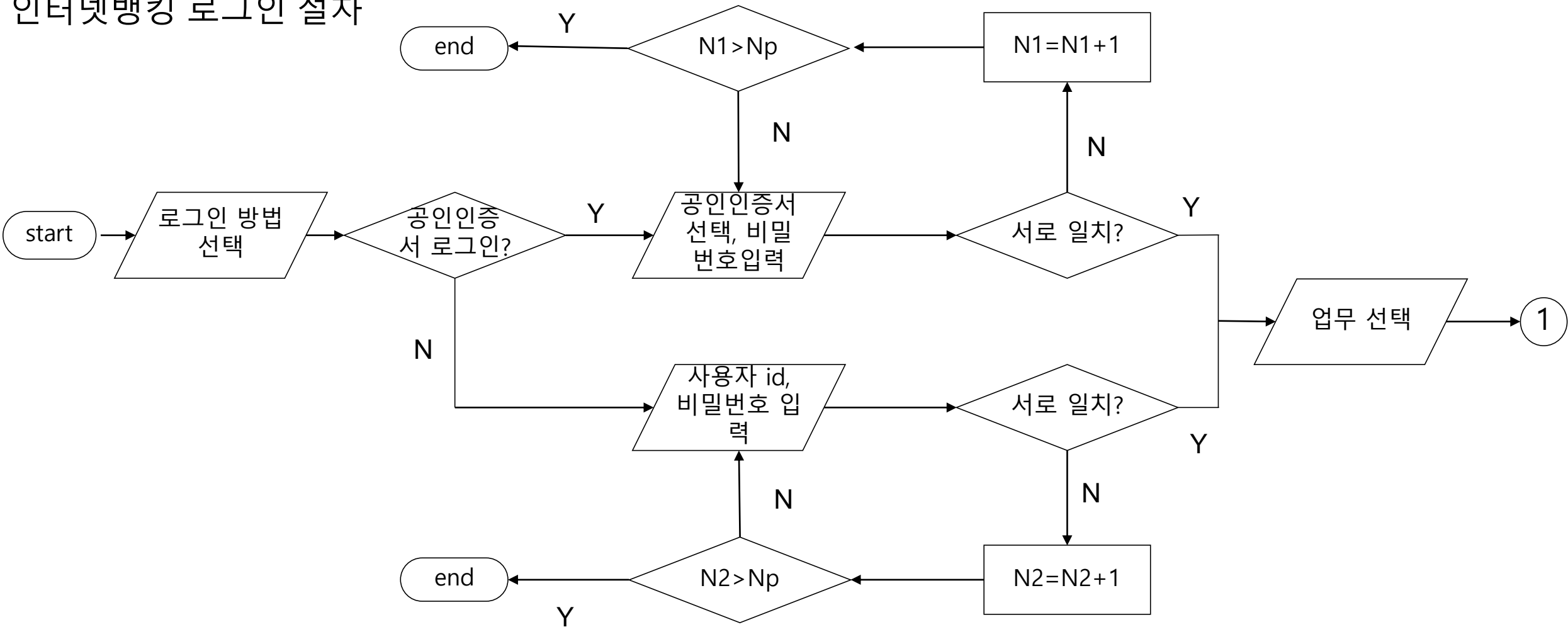
파이썬

```
a=[5,3,8,2,9,4,7,1,0,6]
M=a[0]
for i in range(1,10):
    if a[i] > M:
        M = a[i]
print(M)
```



수학적인 문제가 아닌 경우의 흐름도

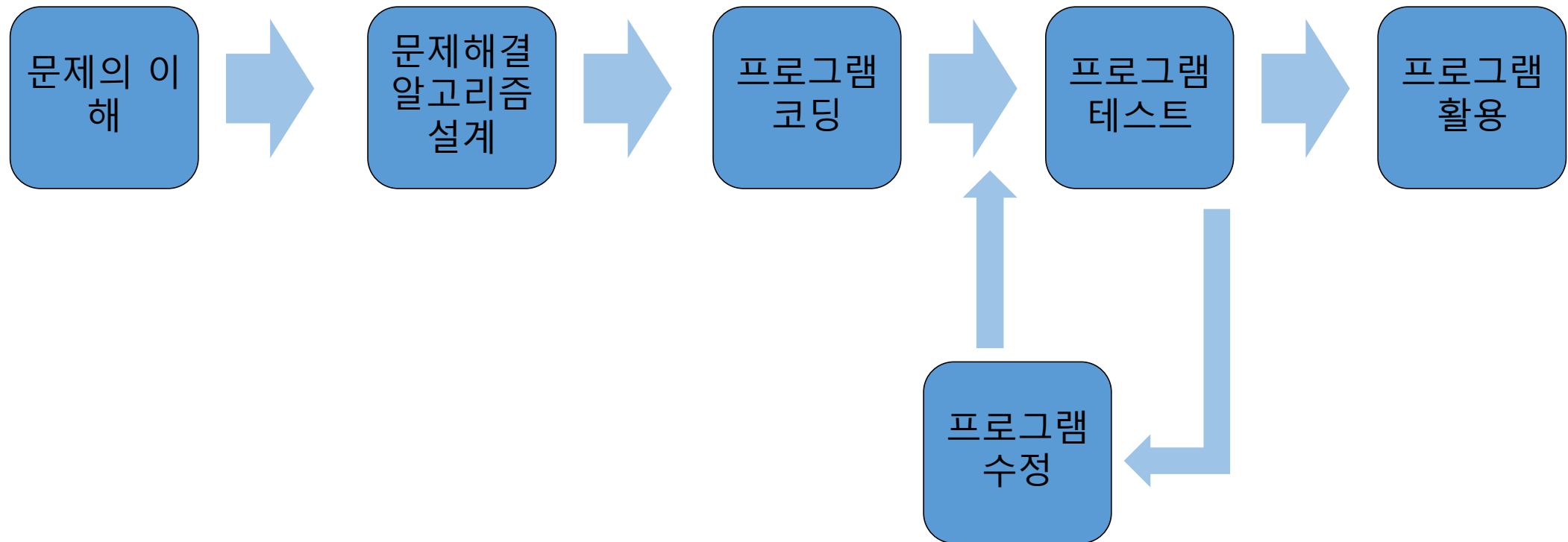
인터넷뱅킹 로그인 절차



✓ N1, N2의 초기값은 0으로 가정



프로그램 작성 단계



문제해결을 위한 알고리즘의 설계 방법

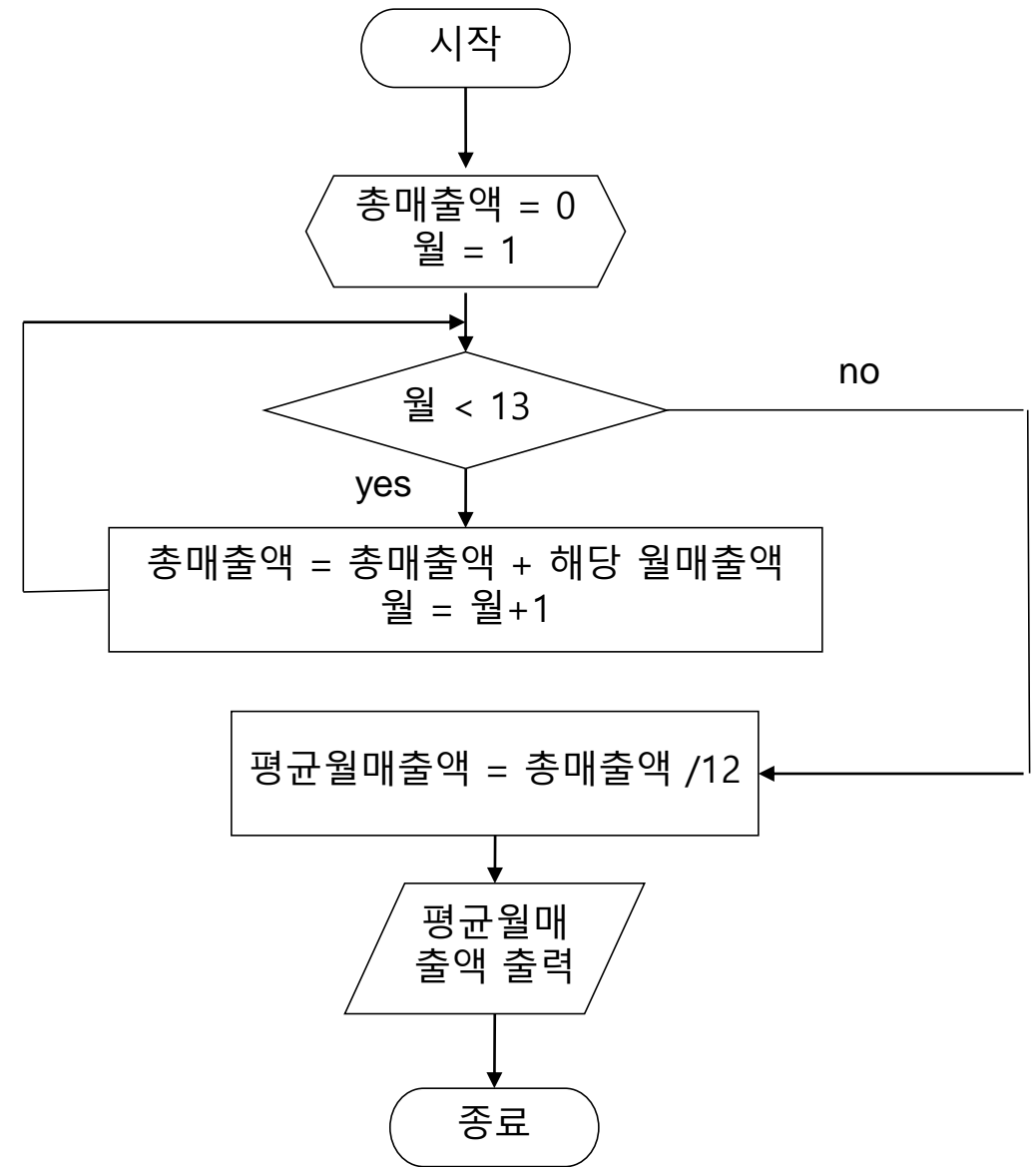
- 의사코드 (pseudocode)
 - ✓ 수행하는 작업들을 프로그래밍 언어와 유사한 방법으로 나타낸 것
- 흐름도
 - ✓ 어떤 문제를 해결하는데 필요한 작업들을 도형들의 관계로 표현
 - ✓ 도형들의 관계가 논리적인 흐름을 나타낸다.
 - ✓ 작업들의 연관 관계와 선후 관계를 시각적으로 보여 줌



작년도 평균 월매출액을 계산한다.

1. 총매출액을 0으로 초기화한다.
2. 1월부터 12월까지의 월매출액을 총매출액에 더한다.
3. 총매출액을 12로 나누어 평균월매출액을 구한다.
4. 평균월매출액을 출력한다.

의사코드(pseudocode)



흐름도(flowchart)



의사코드

Pseudocode



의사코드(pseudocode)

- 의사(疑似)란 '실제와 비슷하다'라는 뜻
- pseudo의 뜻은 '허위의, 가짜의, 모조의'
- 의사코드는 실제와 비슷한, 진짜는 아닌 코드
- 알고리즘의 이해를 쉽게 하기 위한, 단순하면서도 프로그램 코드와 유사한 형식
- 사람이 이해하기 쉬운 형식
- 우리가 보통 사용하는 프로그래밍 언어의 형식과 유사하게 작성



- 프로그램 작성의 직전단계
- 의사코드에 대한 표준은 존재하지 않는다.
- 적당한 들여쓰기(indentation)를 사용한다
- 실제 프로그램의 자세한 사항은 생략

- 내일 할 일을 결정하는
알고리즘의 의사코드

[알고리즘 내일 할 일 결정]

```
if 내일 날씨가 좋으면
    if 야구장 티켓 ≤ 5,000원
        야구장에 간다
    else
        공원에 간다.
else if 내일 날씨가 나쁘면,
    친구에게 전화를 건다
    if 친구가 시간이 있다고 하면,
        같이 저녁을 먹는다.
    else
        혼자 마트에 간다.
else // 날씨가 보통이면
    집에서 쉰다.
```

설명문



- 1부터 100까지의 자연수를 더하는 의사코드

[알고리즘 1부터 100까지 자연수의 합]

```
sum = 0
```

```
n=1
```

```
while (n ≤ 100)
```

```
    sum = sum + n
```

```
    n = n+1
```

```
print sum
```



의사코드 표시 방법

(1) 들여쓰기

- 일반적으로 적당한 들여쓰기(indentation)를 하여, 알고리즘의 구조를 쉽게 파악할 수 있도록 구성한다.

(2) 수학적 표현

- 수학적 의미가 정확한 내용을 수학적 표현으로 기술할 수 있다.
- $\text{for } 1 \leq a \leq 9$ 라는 표현이 가능하다



(3) 단순한 표현

- a와 b의 값을 서로 교환(swap, exchange)하는 작업은 간단히 `swap(a,b)` 또는 `exchange(a,b)`로 표시할 수 있다.

(4) 배열의 인덱스

- 의사코드에서는 배열의 인덱스를 사용하는데 특정한 제한을 받지 않는다.
- 1 또는 다른 자연수부터 시작하는 것으로 설정하여도 무방하다.

(5) 변수 선언

- `number a` 라는 방식으로 변수를 정의해도 무방하다.
- 정수인지 실수인지 특별히 구분할 필요가 없을 경우에는 일반적으로 수라는 표현으로 충분하다.

(6) 변수에 값 저장

- $a=3$, $a:=3$, $a\leftarrow 3$, $a=b+c$ 와 같은 방법을 사용할 수 있다.

(7) 반복 또는 조건이 적용되는 영역

- 반복이 되는 영역 또는 조건이 적용되는 영역을 중괄호, {영역},로 나타낼 수 있다.
- 또는 if ~ fi, for ~ end for, while ~ end while과 같이, 반복 또는 조건을 나타내는 키워드가 적용되는 끝을 명시적으로 표시할 수 있다.

의사코드로 프로그램의 5 가지 기능 표현 방법

(1) 입력

- read, input, get 등의 명령문을 이용하여 입력하는 변수를 나타낸다.

(예) read A

(2) 출력

- write, output, put, print 등의 명령문을 이용하여 출력하는 변수를 나타낸다.

(예) write A

(3) 연산

- 변수에 값을 지정하는 방식으로 연산을 나타낸다. (예) $a=b+c$



(4) 선택 - 조건부 수행

- if, else if, else 등을 조합하여 선택의 조건과 조건부로 수행될 영역을 나타낸다.
- 입력으로 학년의 정보를 받아, 적당한 출력을 만드는 의사코드다.

[알고리즘 학년 변환]

```
입력 hakyun
출력 학년의 영문명
get hakyun
if hakyun == 4
    print "Senior"
else if hakyun == 3
    print "Junior"
else if hakyun == 2
    print "Sophomore"
else if hakyun == 1
    print "Freshman"
else print "Error"
```



- ✓내일 비가 오면, 집에서 쉰다.
- ✓월급이 10% 인상된다면, 적금에 가입한다.
- ✓운전면허시험에 합격한다면, 축하 행사를 한다.
- ✓나이가 60세 이상이면,
연금을 받는다.
- 나이가 20세 이하이면,
지원금을 받는다.



(5) 반복

- while, for, repeat 명령문을 이용하여 표현한다.
- 입력으로 받은 정수들을 누적하는 알고리즘의 의사코드
 - ✓ 알고리즘은 n 의 값이 0일 때 종료하고, 그 때의 sum 값을 출력한다.

[알고리즘 입력 정수들의 합]

```
number sum, n
```

```
sum = 0
```

```
read n
```

```
while ( $n \neq 0$ )
```

```
    sum = sum + n
```

```
    read n
```

```
print sum
```



- 배열명 a에 저장된 5개의 데이터, 배열의 인덱스는 0부터 4라고 가정할 때, 배열에 저장된 숫자의 합을 구하는 의사코드

[알고리즘 배열의 합]

number a[5]={4,3,8,6,7}, k

sum = 0

for k in 0...4

 sum = sum + a[k]

print sum



- 1부터 100까지의 합을 구하는 의사코드

[알고리즘 1부터 100까지의 합]

```
sum = 0  
n=1  
while (n ≤ 100)  
    sum = sum + n  
    n = n+1  
print sum
```



(예 1) 두 정수 중 큰 수를 찾는다.

[알고리즘 큰 수 찾기]

입력: 두 숫자 a, b

출력: 큰 숫자

if (a > b)

 print a

else if (a < b)

 print b

else

 print "두 수는 같다"



(예 2) 1과 자신의 수는 제외한 자연수의 약수를 출력한다.

[알고리즘 약수 찾기]

입력: 숫자 a

출력: a 의 약수

$b=2$

while ($b \leq a/2$) {

 if a/b 의 나머지가 0이면

 print b

$b=b+1$

}



[연습문제 1] 정수로 표시된 4과목의 평균을 구하는 알고리즘의 의사코드를 작성하라.

[연습문제 2] 다음 학기 수강 신청할 과목들을 결정하는 알고리즘의 의사코드를 작성하라.

[연습문제 3] 10부터 100사이의 자연수 하나를 입력 받는다. n 이라 한다. n 의 약수들의 합을 구하는 알고리즘의 의사코드를 작성하라.

[연습문제 4] 인터넷 뱅킹에서 계좌이체 절차의 의사코드를 작성하라.



흐름도

Flowchart

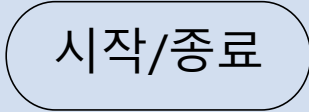
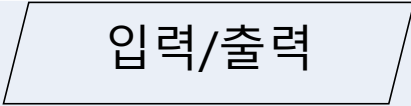
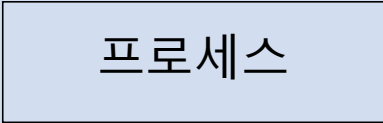
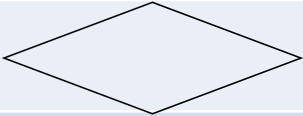
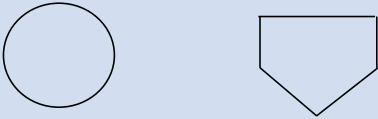

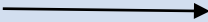


흐름도 란

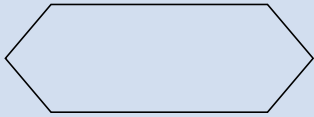
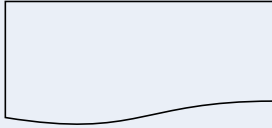

- ✓ 어떤 문제를 해결하는데 필요한 작업들을 도형들의 관계로 표현
- ✓ 도형들의 관계가 논리적인 흐름을 나타낸다.
- ✓ 작업들의 연관 관계와 선후 관계를 시각적으로 보여 줌

3. 흐름도

흐름도에서 사용되는 표시

심벌 이름	도형	내용
터미널 심벌		흐름도의 시작과 끝을 나타내는 기호
입출력 심벌		입출력 작업 표시
프로세스 심벌		연산 명령문 등 처리해야 할 작업 내용
판단 심벌		판단을 나타낸다
연결심벌		페이지 내, 외의 연결을 나타낸다
미리 정의된 프로세스 심벌		모듈, 함수, 메소드, 하위절차
흐름심벌		연결 흐름 표시

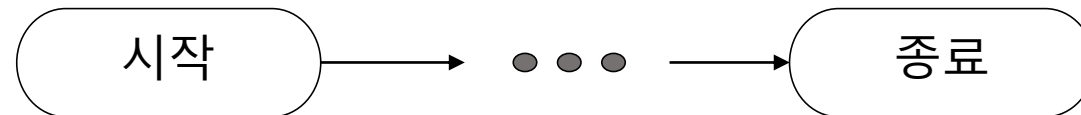


심벌 이름	도형	내용
준비 심벌		변수의 초기화
문서출력 심벌		문서로 출력
데이터베이스 심벌		데이터베이스

- 터미널(terminal) 심벌

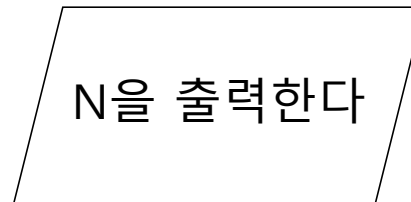
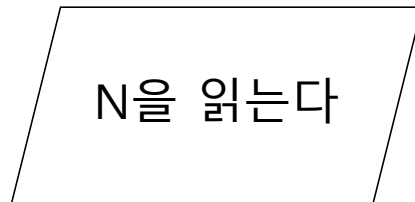
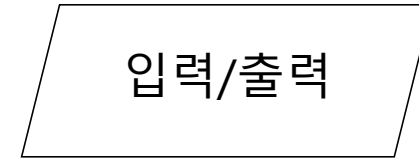
- 모서리가 원형인 사각형(rounded rectangle)으로 표시한다.
- 흐름도의 시작과 끝을 나타내는 도형이다.
- 시작과 종료로 표시한다.

시작/종료



- 입출력 심벌

- 데이터의 입출력 작업 표시
- 평행사변형 (parallelogram)으로 표시

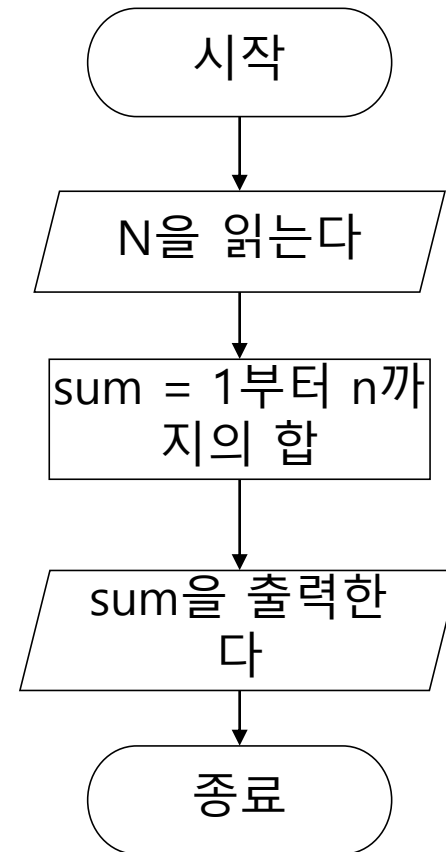


- 프로세스 심벌(process symbol)

프로세스

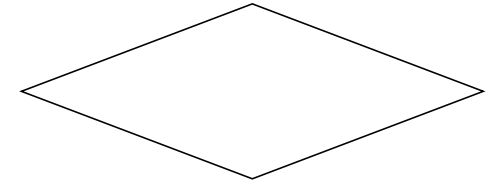
- 수학적 연산 또는 데이터 처리를 표시한다.
- 직사각형(rectangle)으로 표시

두 수 a,b값을
교환한다

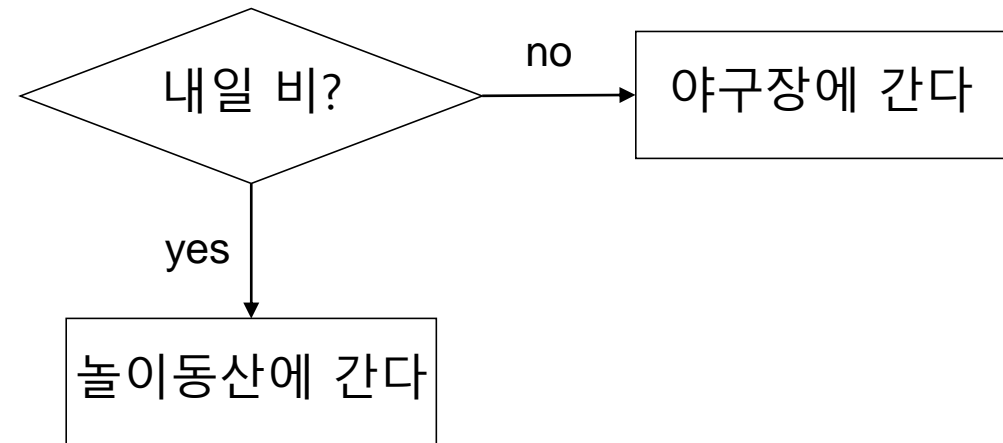


- 판단(decision)

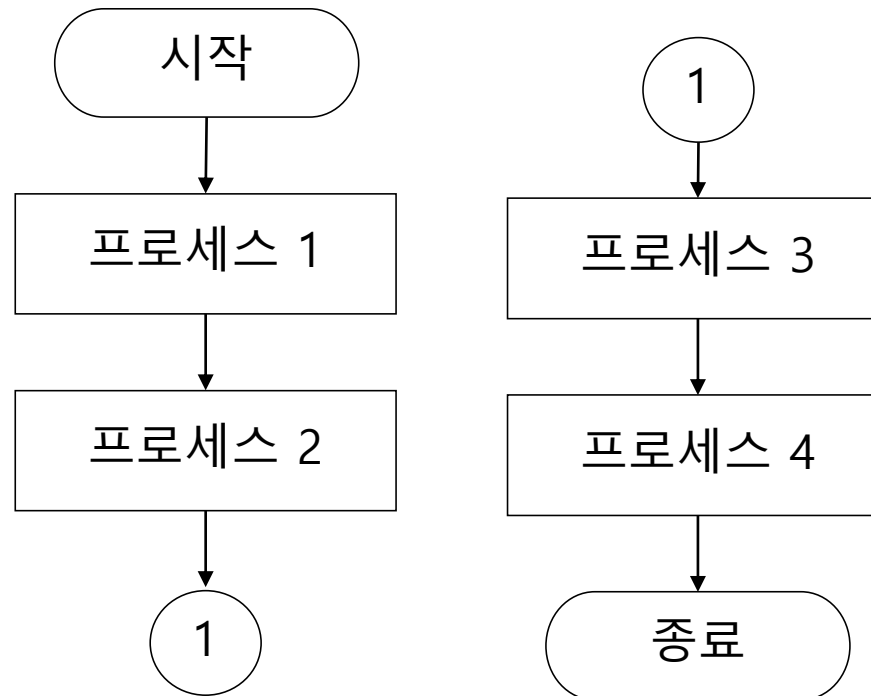
- 조건을 판단하여 결과는 예/아니오 또는 참/거짓이 된다.
- 다이아몬드로 표시한다.



```
if( 내일 비가 오면) then
    놀이동산에 간다
else
    야구장에 간다
```

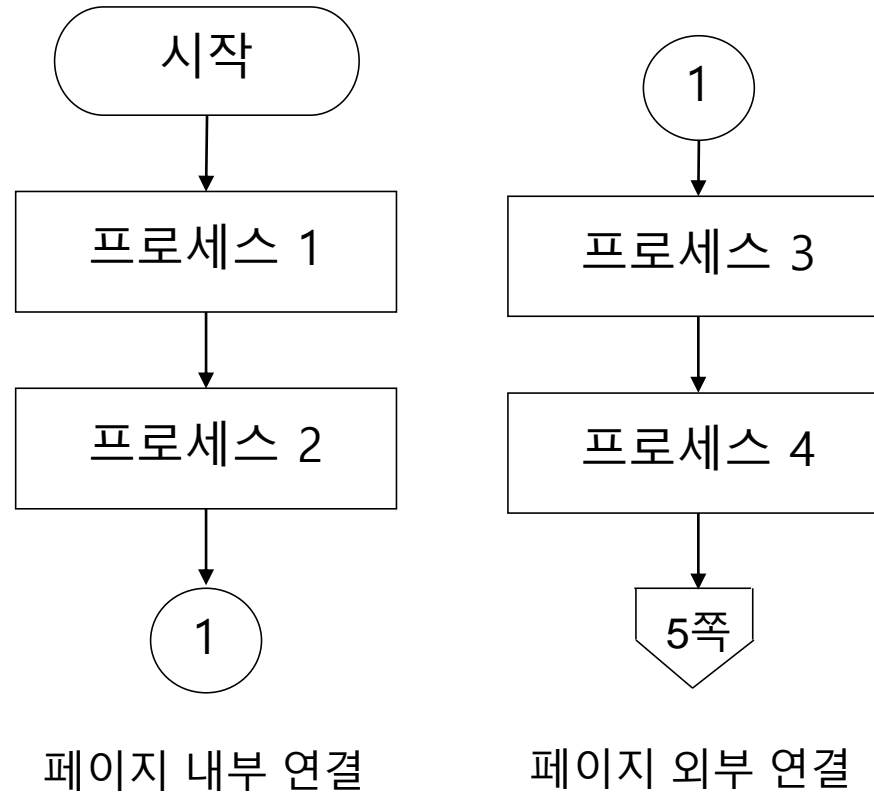


- 페이지 내 연결(connector) 심벌
 - 페이지 내의 위치로 이동하는 것을 표시한다.
 - 원으로 표시

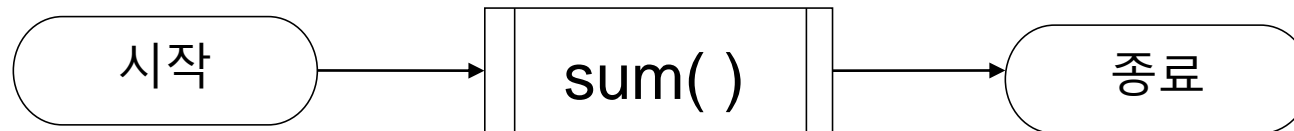
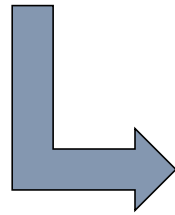
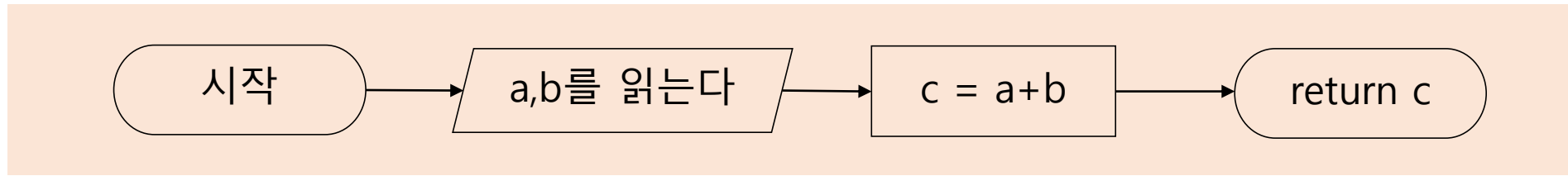


페이지 내부 연결

- 페이지 외부로 연결(connector) 심벌
 - 야구의 홈플레이트 도형으로 표시한다.
 - 페이지 외부의 위치로 이동하는 것을 표시한다.

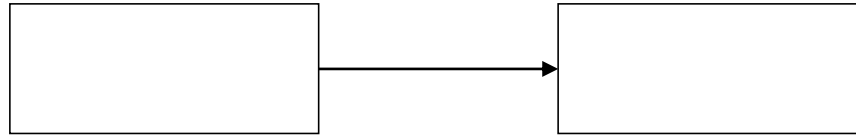


- 미리 정의된 프로세스 심벌 – 모듈, 함수, 메소드, 하위절차

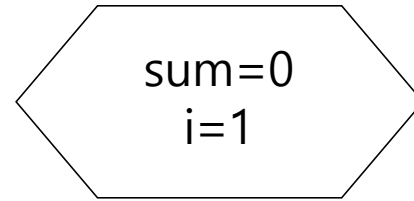


- 흐름 (flow)

- 화살표로 표시
- 작업의 선후관계를 표시



- 준비 (preparation)
 - 육각형으로 표시
 - 변수의 초기화 표시



- 문서출력 (document)
 - 종이로 출력하는 내용을 표시

이름, 학번

- 데이터베이스

- 원통형으로 표시한다.
- 데이터의 저장소, 즉 데이터베이스를 표시한다.



흐름도 작성 원칙

(1) 좌 → 우, 상 → 하



(2) 하나의 도형은 하나의 작업을 나타낸다

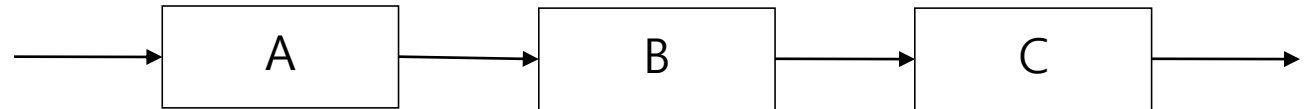
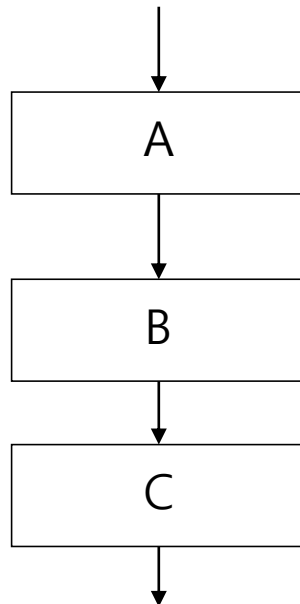


흐름도의 구조

- 기본구조: 순서 표시
- 선택의 형태
- 반복의 형태
- 순서, 선택, 반복의 형태를 적당히 활용하여 모든 흐름도의 구조를 완성

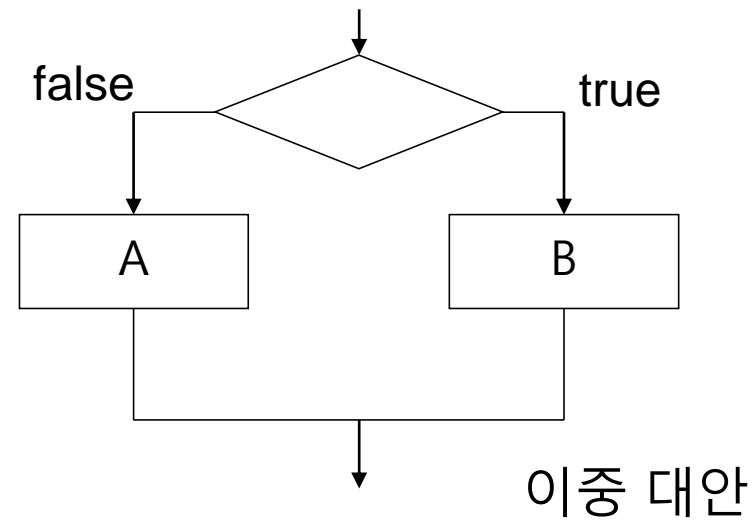
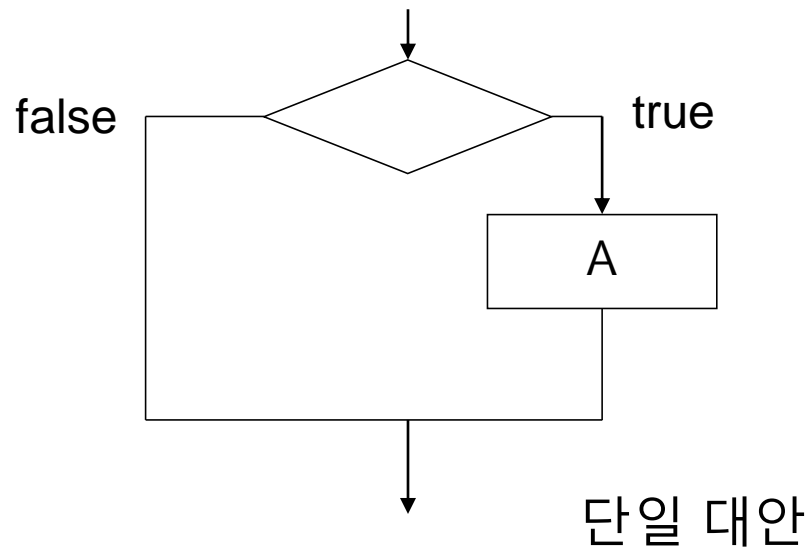
● 순서 (sequence)를 나타내는 방법

- 프로그램의 모든 작업은 순차적으로 진행한다(병렬처리인 경우는 예외).
- 그 순서를 나타내는 방법은 단위 작업들을 하나의 순서로 나열하는 것이다.
- 다음은 작업 A, B, C가 순서대로 진행되는 흐름도
- B는 A가 종료된 후 시작. 작업 A, B, C는 모두 종료되어야 한다. B에서 중단될 수 없다.

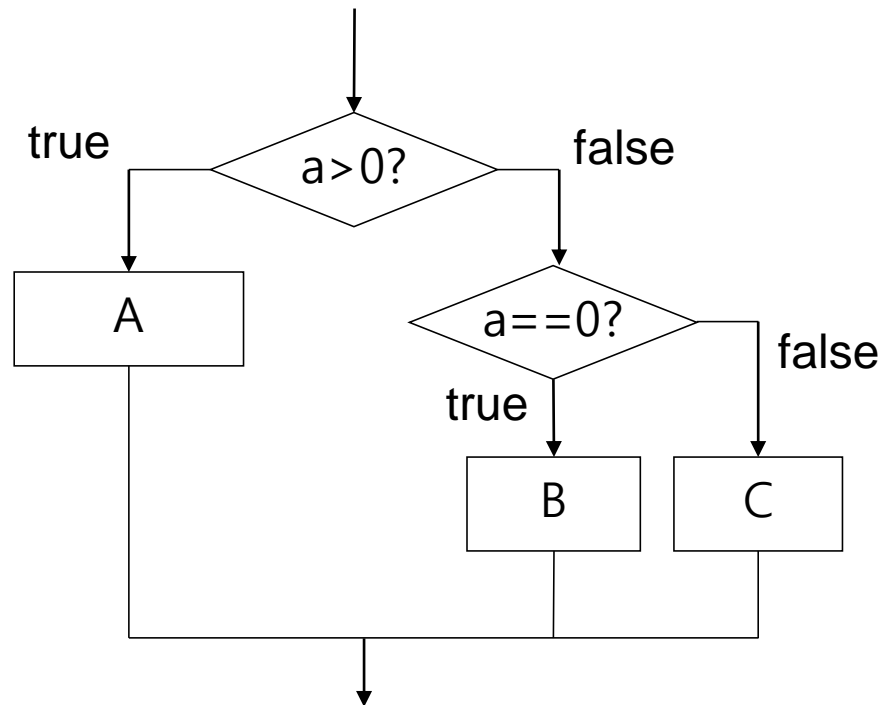


● 선택을 나타내는 방법

- 조건의 판단에 따라 둘 중 하나의 경로로만 이동
- 단일 대안
- 이중 대안
- 삼중 대안

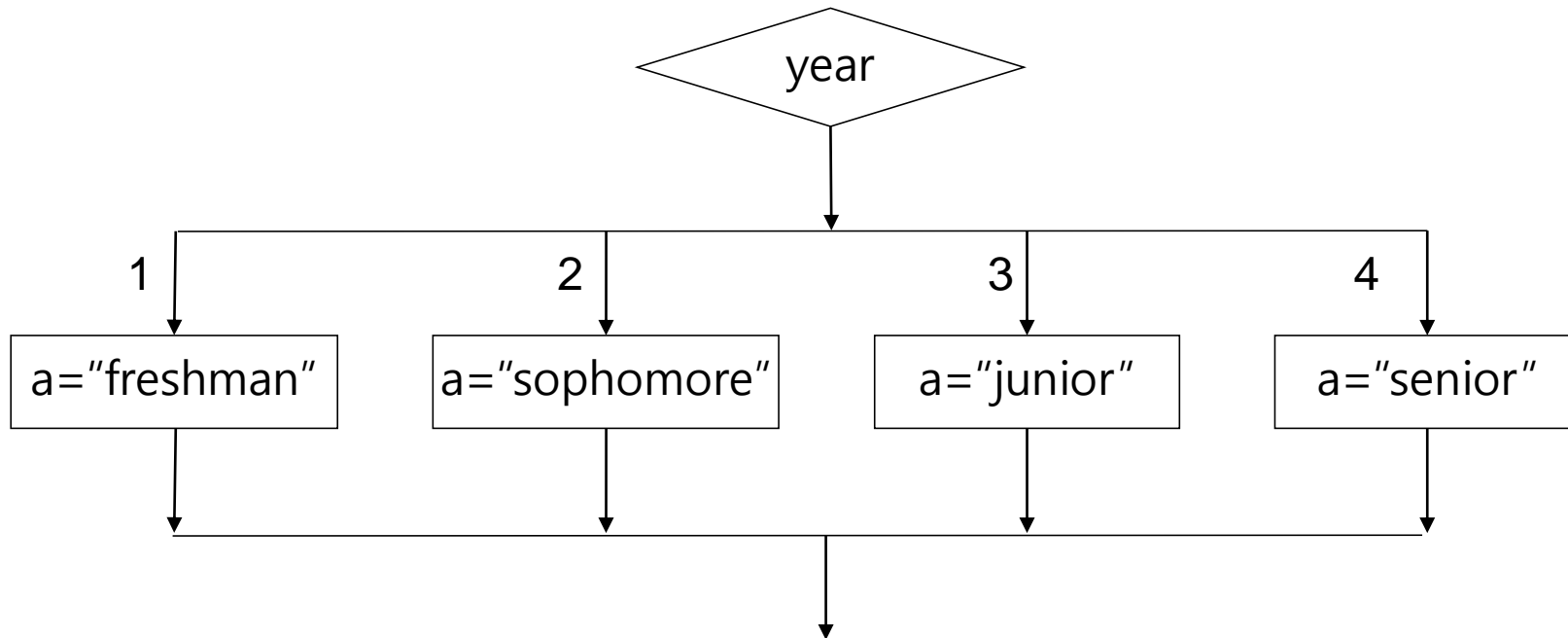


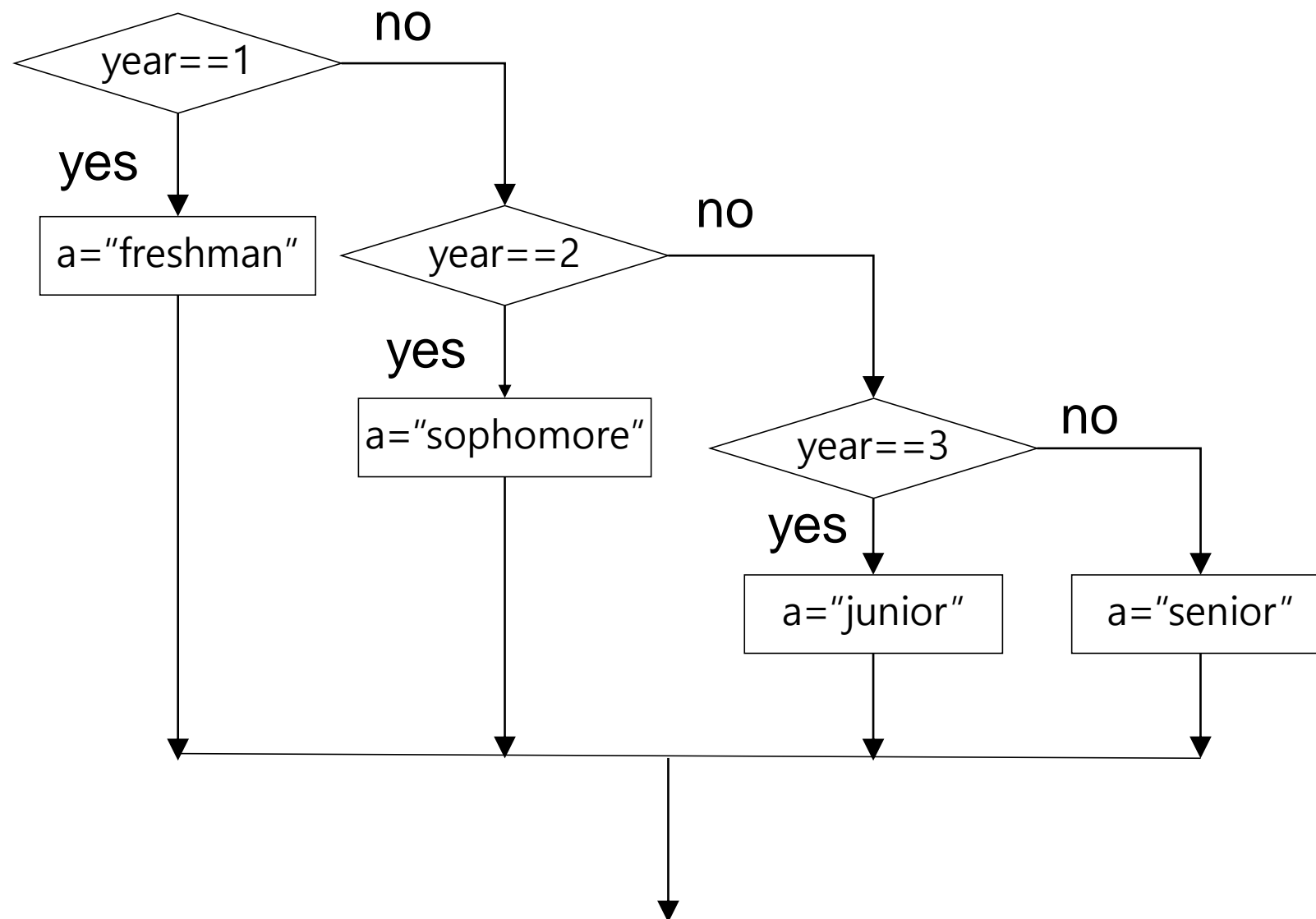
삼중 대안



case 형태의 흐름도

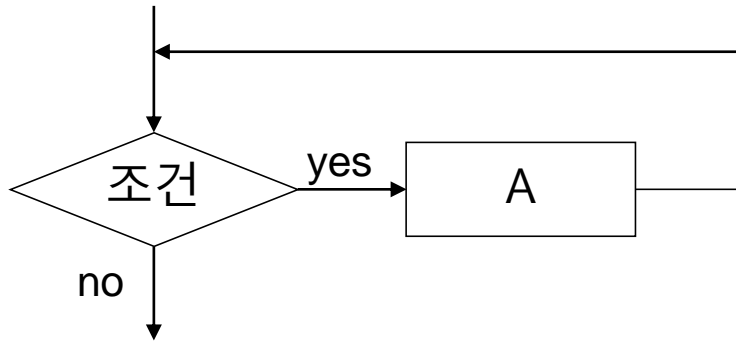
- 변수의 값에 따라 작업이 결정됨
- 변수는 정수



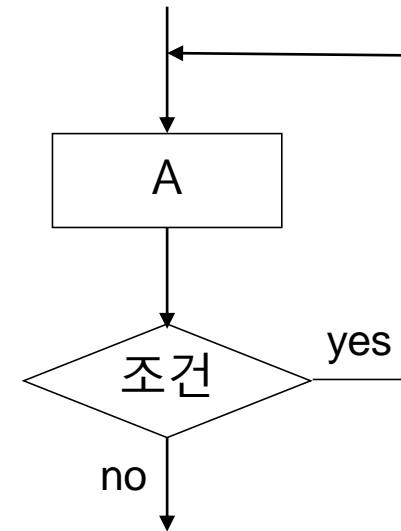


- 반복(loop, iteration, repetition)을 나타내는 방법

- 정해진 숫자만큼 또는 어떤 조건을 만족하는 동안 동일한 작업을 반복하는 형태
- 판단과 결합하여 사용



판단 후 반복 수행



판단 전 반복 수행



- 1부터 100까지의 자연수를 더하는 알고리즘의 흐름도

[알고리즘] 1부터 100까지 자연수의 합

```
sum = 0
```

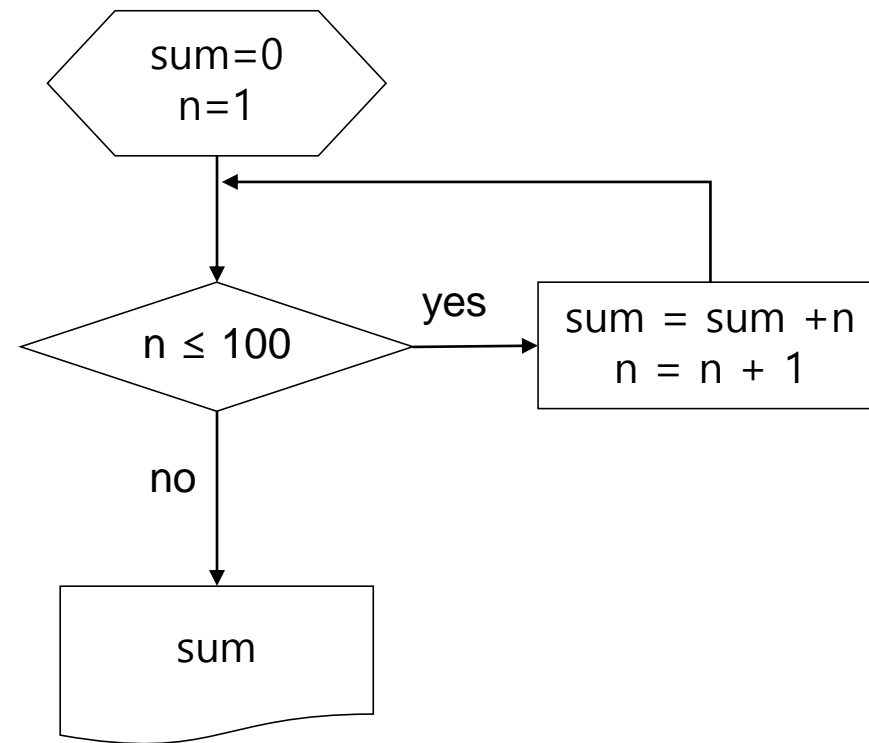
```
n=1
```

```
while (n ≤ 100)
```

```
    sum = sum + n
```

```
    n = n+1
```

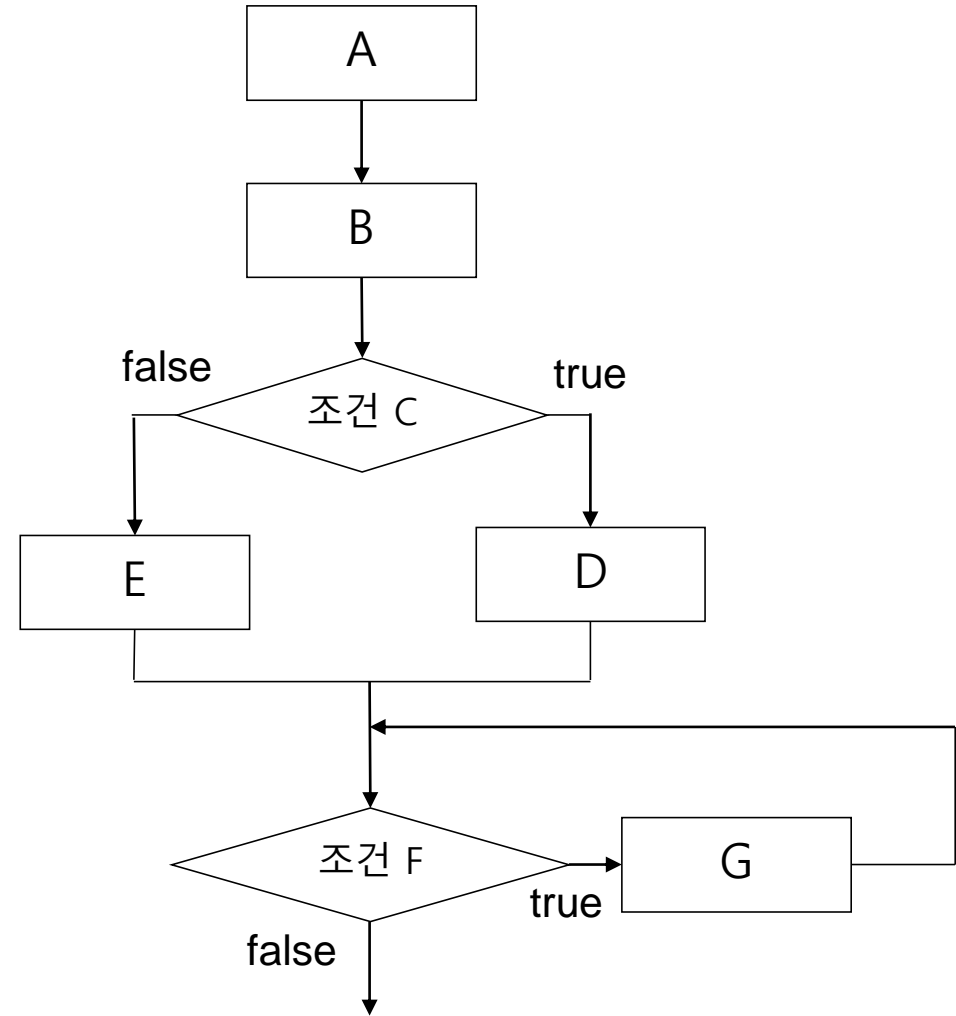
```
print sum
```



의사코드와 흐름도

- 판단과 반복의 형태 포함

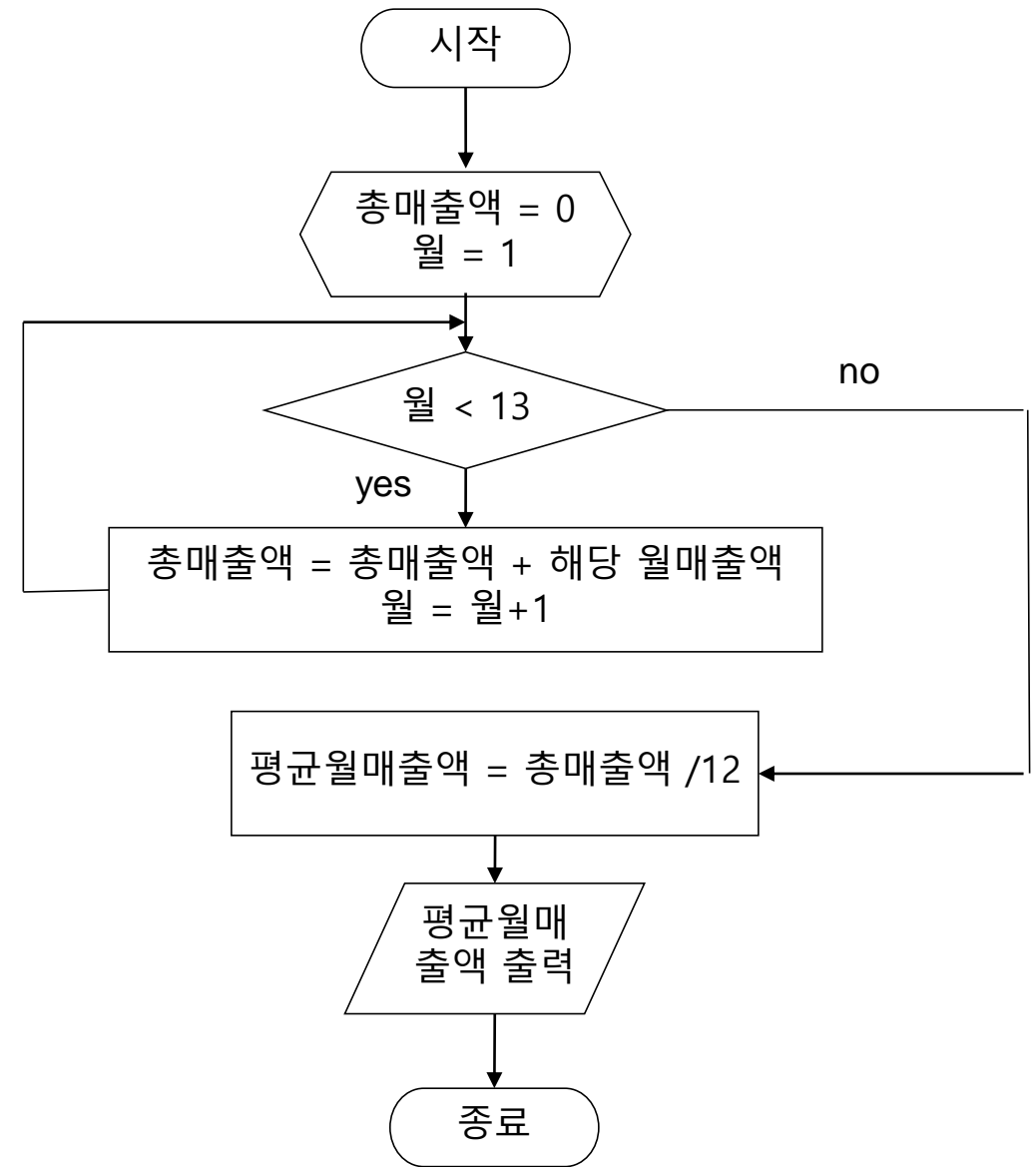
```
A
B
if 조건C is true then
    D
else
    E
endif
while 조건F is true
    G
end while
```



작년도 평균 월매출액을 계산한다.

1. 총매출액을 0으로 초기화한다.
2. 1월부터 12월까지의 월매출액을 총매출액에 더한다.
3. 총매출액을 12로 나누어 평균월매출액을 구한다.
4. 평균월매출액을 출력한다.

의사코드(pseudocode)



흐름도(flowchart)



[연습문제 1] 정수로 표시된 4과목의 평균을 구하는 알고리즘의 흐름도를 작성하라.

[연습문제 2] 다음 학기 수강 신청할 과목들을 결정하는 알고리즘의 흐름도를 작성하라.

[연습문제 3] 10부터 100사이의 자연수 하나를 입력 받는다. n 이라 한다. n 의 약수들의 합을 구하는 알고리즘의 흐름도를 작성하라.

[연습문제 4] 인터넷 뱅킹에서 계좌이체 절차의 흐름도를 작성하라.



4주차 강의 요약

- 알고리즘 소개
- 알고리즘의 표현 방법
- 의사코드
- 흐름도



4주차 끝

수고하셨습니다.