

# Chapter 02

## 알고리즘 작성과 표현



# 목차

1. 알고리즘의 개요
2. 일상생활 속 알고리즘
3. 수학 속 알고리즘
4. 순서도와 의사코드

# 학습목표

- 알고리즘의 의미와 3가지 제어 구조를 알아봅니다.
- 일상생활 속 문제 해결 과정을 알고리즘으로 작성해봅니다.
- 수학 문제를 알고리즘으로 작성해봅니다.
- 순서도와 의사코드 작성 방법을 알아봅니다.

# Section 01

## 알고리즘의 개요



## ■ 알고리즘의 개념

- 특정 업무를 수행하기 위한 절차 또는 명령어의 집합
- 주어진 문제를 해결하는 데 필요한 방법과 절차가 순서대로 기술한 것
- 즉 우리가 평소 사용하는 컴퓨터 프로그램은 알고리즘을 프로그래밍 언어를 이용하여 코드로 변환한 것을 의미함



그림 2-1 알고리즘과 프로그램의 관계



## ■ ‘야채 계란말이’ 알고리즘의 예시

- ① 냉장고에서 ‘야채’와 ‘계란’을 꺼낸다.
- ② 야채를 깨끗이 씻고 잘게 다진다.
- ③ 계란과 야채를 그릇에 넣고 섞는다.
- ④ 가스 불을 켜다.
- ⑤ 후라이팬에 기름을 두른다.
- ⑥ 계란 물은 부은 후 어느 정도 익으면 돌돌 만든다.
- ⑦ 계란말이가 완성되면 먹기 좋게 자른 후 그릇에 담는다.

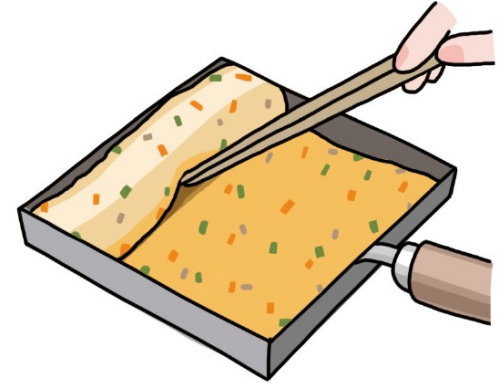


그림 2-2 야채 계란말이 알고리즘



## ■ 더 완벽한 ‘야채 계란말이’ 알고리즘의 예시

- ① 냉장고에 ‘야채’와 ‘계란’이 있으면 꺼낸다.
- ② 만약 재료가 없다면 마트에 가서 구입한다.
  - ②-1 시간적 여유가 있다면 저렴하게 구입할 수 있는 대형 마트에서 구입한다.
  - ②-2 시간적 여유가 없다면 집에서 가까운 슈퍼에서 구입한다.
- ③ 야채를 깨끗이 씻고 잘게 다진다.
- ④ 계란과 야채를 그릇에 넣고 섞는다.
- ⑤ 가스 불을 켜다.
- ⑥ 후라이팬에 기름을 두른다.
- ⑦ 계란 물은 부은 후 어느 정도 익으면 돌돌 만든다.
- ⑧ 계란말이가 완성되면 먹기 좋게 자른 후 그릇에 담는다.
- ⑨ 케첩을 좋아하면 뿌리고 싫어하면 생략한다.

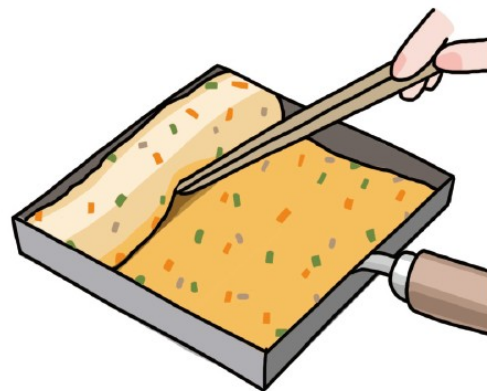


그림 2-3 수정한 야채 계란말이 알고리즘



## ■ 알고리즘 설계란

- 알고리즘을 설계할 때는 제어 구조를 이용함
- 제어 구조는 알고리즘에서 명령의 실행 순서를 결정하는 구조로 순차 구조, 선택 구조, 반복 구조가 있음

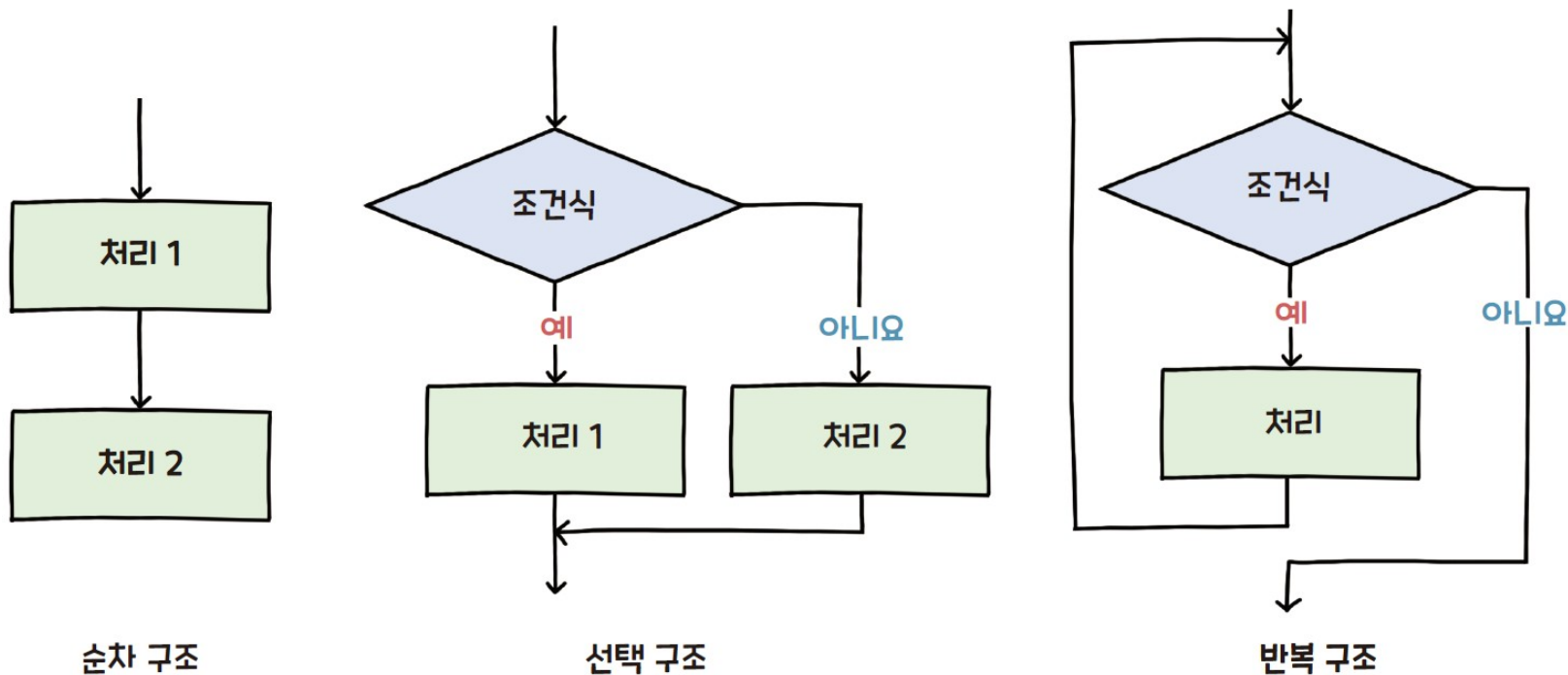


그림 2-4 제어 구조





## ■ 알고리즘 설계란

- 순차 구조 : 문제를 해결하기 위해 가장 효율적인 방법을 찾아내는 과정
- 선택 구조 : 특정 조건을 만족하는지 여부에 따라 다음 명령을 선택하여 실행
- 반복 구조 : 어떤 문제를 해결하기 위한 같은 동작을 반복하여 실행
- 기차 탑승 과정의 제어 구조 예시

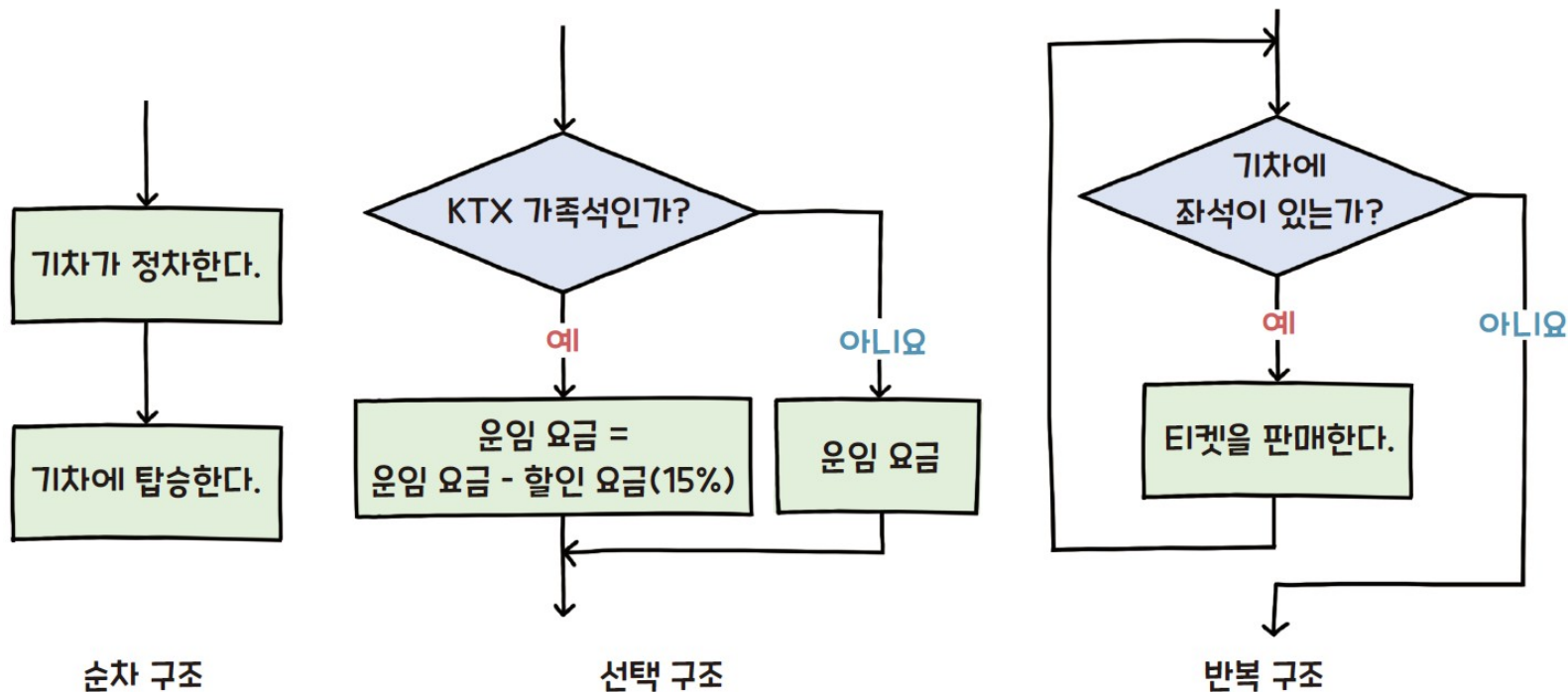


그림 2-5 제어 구조의 예(© 중학교 정보 교과서, 2018, 한빛아카데미)

## Section 02

# 일상생활 속 알고리즘



## ■ 아침에 발생할 수 있는 여러 상황을 고려한 등교 알고리즘

- ① 알람 소리에 기상한다.
- ② 아침 식사는 간단한 빵 종류로 해결한다.
- ③ 양치질과 세수를 한다.
- ④ 날씨 정보를 참고해서 옷을 고른다.
- ⑤ 신발을 신고 자전거를 꺼낸다.
- ⑥ 안전모를 착용하고 자전거 전용 도로를 이용해서 학교로 향한다.
- ⑦ 학교에 도착하면 자전거를 자전거 거치대에 세운 후 학급으로 향한다.



그림 2-6 등교 알고리즘



## 확인문제

위의 등교 알고리즘에서 지각을 하지 않기 위한 과정을 좀 더 명확한 알고리즘으로 수정하십시오.

### 정답

- ① 알람 소리에 기상한다.
- ② 시간이 있으면 식사를 하고 없으면 간식을 챙긴다.
- ③ 양치질과 세수를 한다.
- ④ 날씨 정보를 참고해서 옷을 고른다.
- ⑤ 신발을 신고 자전거를 꺼낸다.
- ⑥ 안전모를 착용하고 자전거 전용 도로를 이용해서 학교로 향한다.
- ⑦ 시계를 보고 시간이 충분하면 천천히 타고, 충분하지 않으면 페달을 밟아 빨리 간다.
- ⑧ 학교에 도착하면 자전거를 자전거 거치대에 세운 후 학급으로 향한다.



## ■ 키오스크를 이용한 메뉴 주문 알고리즘

- ① 키오스크에서 메뉴를 검색한다.
- ② 단품과 세트 메뉴 중 원하는 메뉴를 선택한다.
- ③ 배가 많이 고프면 사이드 메뉴를 선택하고, 그렇지 않으면 생략한다.
- ④ 결제할 금액을 확인하고 카드 투입구에 카드를 넣는다.
- ⑤ 결제가 완료되면 주문 번호와 영수증을 챙긴다.
- ⑥ 주문 번호가 호출되면 음식을 수령한다.

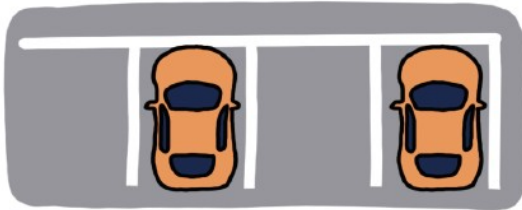


그림 2-7 키오스크 주문 알고리즘

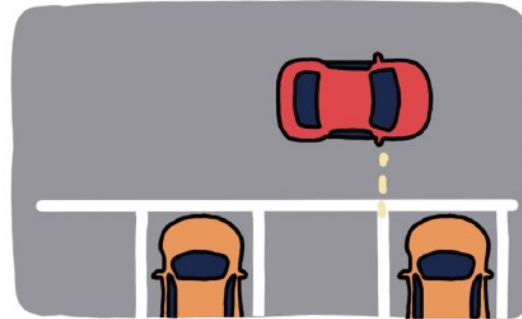


## ■ 아파트 주차장에서 후면 주차하는 알고리즘

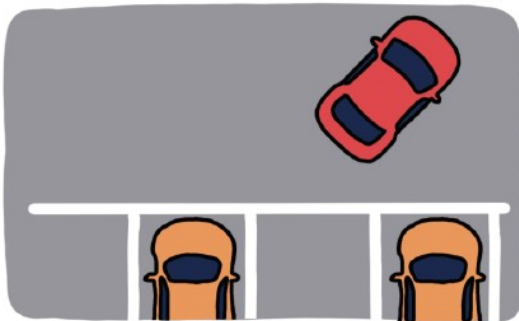
① 빈자리를 찾는다. 이때 가능하면 집에서 가까운 자리로 찾는다.



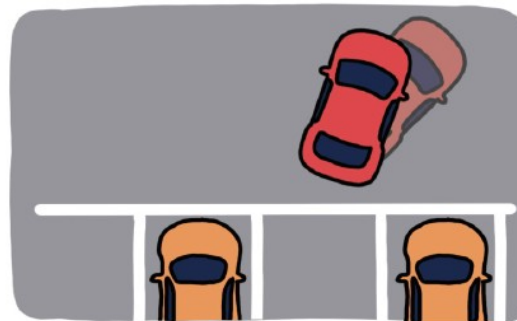
② 주차 자리를 찾았다면, 사이드미러와 우측 라인을 맞춘다.



③ 핸들을 왼쪽으로 돌려 11시 방향으로 전진한다.



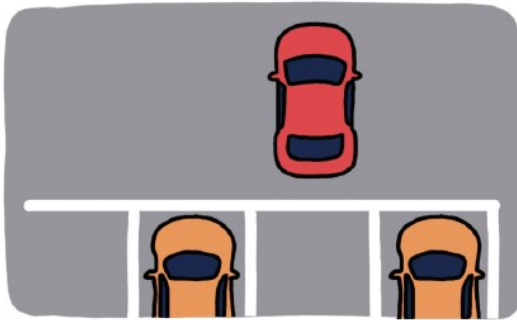
④ 후진 기어를 넣고, 핸들을 오른쪽으로 끝까지 돌려 후진한다.





## ■ 아파트 주차장에서 후면 주차하는 알고리즘

⑤ 사이드미러를 통해서 주차 선에 자동차를 반듯하게 정렬한다.



⑥ 파킹 기어를 넣고 차에서 내린다. 이때 옆 차에 피해를 주지 않도록 조심한다.

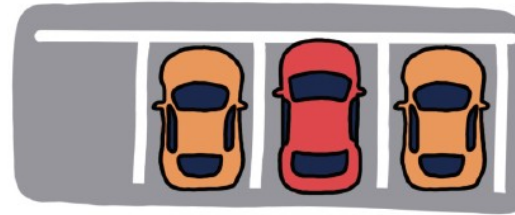


그림 2-8 후면 주차 알고리즘

# Section 03

## 수학 알고리즘





## ■ 생년월일에 따른 만 나이 계산 알고리즘

표 2-1 만 나이 계산 알고리즘

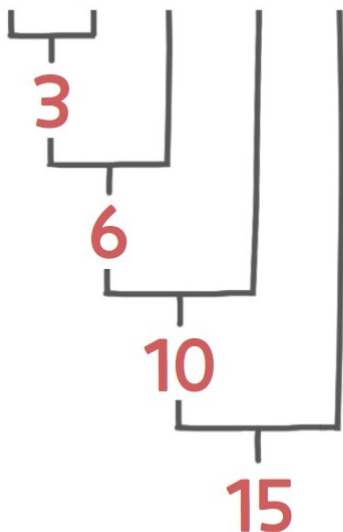
| 출생일           | 알고리즘   | 결과       |
|---------------|--|----------|
| 1974년 1월 20일  | ① 기준년도에서 출생 연도 빼기<br>$(2020 - 1974 = 46)$<br>② 1월 20일은 11월 5일 이전이므로 추가 뺄셈 없음                     | 만 나이 46세 |
| 1974년 12월 10일 | ① 기준년도에서 출생 연도 빼기<br>$(2020 - 1974 = 46)$<br>② 12월 10일은 11월 5일 이후이므로 추가로 1 빼기<br>$(46 - 1 = 45)$ | 만 나이 45세 |
| 1976년 2월 22일  | ① 기준년도에서 출생 연도 빼기<br>$(2020 - 1976 = 44)$<br>② 2월 22일은 11월 5일 이전이므로 추가 뺄셈 없음                     | 만 나이 44세 |
| 1976년 11월 6일  | ① 기준년도에서 출생 연도 빼기<br>$(2020 - 1976 = 44)$<br>② 11월 6일은 11월 5일 이후이므로 추가로 1 빼기<br>$(44 - 1 = 43)$  | 만 나이 43세 |
| 1984년 6월 20일  | ① 기준년도에서 출생 연도 빼기<br>$(2020 - 1984 = 36)$<br>② 6월 20일은 11월 5일 이전이므로 추가 뺄셈 없음                     | 만 나이 36세 |

# 1부터 100까지 정수의 합 구하기



## ■ 방법1 : 순서대로 계산하기

$$1+2+3+4+5+6+7+ \dots +98+99+100$$



이 방법은 시간이 많이 소요되고  
연산을 잘못할 수도 있습니다.

그림 2-9 1부터 100까지 정수의 합을 구하는 방법

# 1부터 100까지 정수의 합 구하기



## ■ 방법2 : 알고리즘으로 만들기

### ① 규칙 발견하기

$$1+2+3+4+5+6+ \cdots +96+97+98+99+100$$

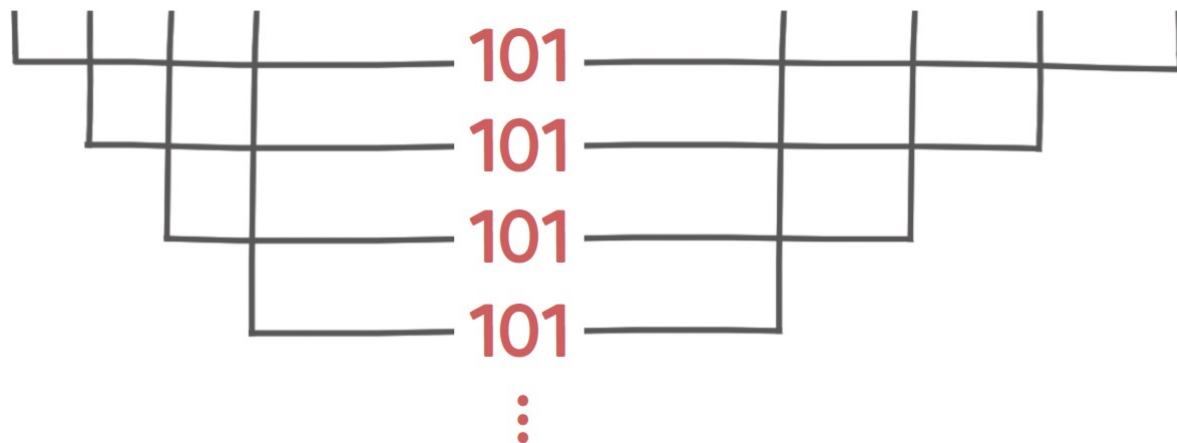


그림 2-10 규칙 발견

# 1부터 100까지 정수의 합 구하기



## ■ 방법2 : 알고리즘으로 만들기

### ② 공식 만들기

$$(1 + \text{마지막 정수}) \times (\text{마지막 정수} / 2)$$

그림 2-11 공식 만들기

### ③ 공식 적용하기

$$(1 + 100) \times (100 / 2) = 101 \times 50 = 5050$$

그림 2-12 공식 적용

# 색칠한 부분의 넓이 구하기



■ 문제 : 색칠한 부분의 넓이 구하기

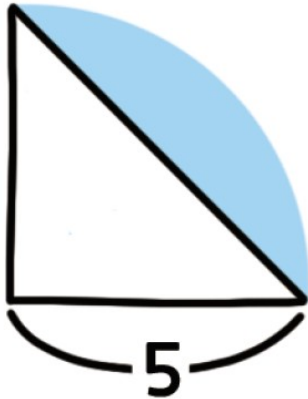


그림 2-13 문제

# 색칠한 부분의 넓이 구하기



## ■ 문제 해결

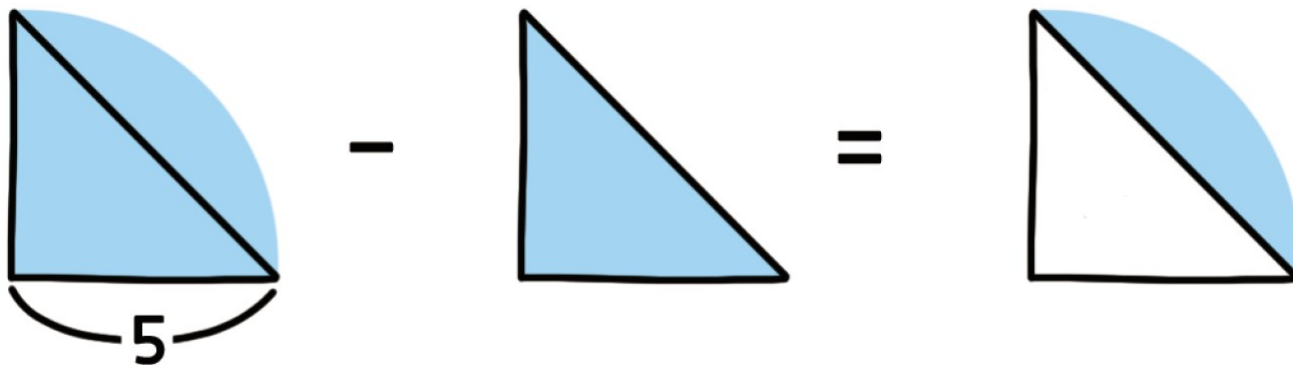


그림 2-14 문제 해결 방법

## ■ 문제 해결 공식

$$((\pi \times 5 \times 5) \div 4) - ((5 \times 5) \div 2) = \frac{(25\pi - 50)}{4}$$

그림 2-15 색칠한 부분의 넓이



- 문제 : 물건의 금액을 계산하여 동전의 개수를 가장 적게 거슬러주기  
(단 동전은 100원, 10원, 1원을 이용함)

- 총 구매 금액 : 1,532원
- 받은 돈 : 2,000원
- 거스름돈 : 2,000원 - 1,532원 = 486원

- 문제 해결

- 100원 4개, 10원 6개, 1원 8개 = 486원

$$\left( \text{100} \times 4 \right) + \left( \text{10} \times 6 \right) + \left( \text{1} \times 8 \right)$$

그림 2-16 동전의 개수를 최소화한 거스름 돈



## ■ 문제 해결 알고리즘

### 100원짜리 개수 구하기

468원에서 100원 미만이 될 때까지 100을 반복해서 뺀다.

이때 반복 횟수가 100원짜리 동전의 개수이다.



### 10원짜리 개수 구하기

68원에서 10원 미만이 될 때까지 10을 반복해서 뺀다.

이때 반복 횟수가 10원짜리 동전의 개수이다.



### 1원짜리 개수 구하기

8원에서 0원까지 1을 반복해서 뺀다.

이때 반복 횟수가 1원짜리 동전의 개수이다.



100원 반복 횟수, 10원 반복 횟수, 1원 반복 횟수가 각각 거스름돈의 동전 개수이다.

그림 2-17 거스름돈 알고리즘



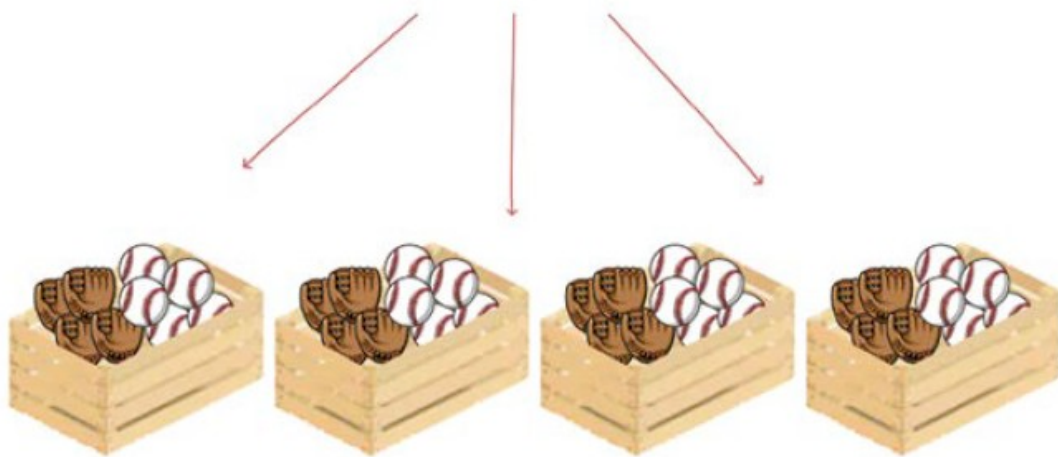


## 확인문제

크리스마스 선물로 아이들에게 야구 글러브와 공을 선물하려고 한다. 글러브와 공의 개수가 각각 48개와 60개라고 할 때, 최대한 많은 아이들에게 나눠주기 위해 필요한 상자의 개수를 구하시오. 단 각 상자에 들어가는 글러브와 공의 개수는 모두 같아야 하고, 글러브와 공을 남기면 안 된다.

**글러브 48개, 야구공 60개**

- 최대한 많은 학생들에게
- 남김없이 나누어 줄 것
- 똑같이 나누어 줄 것






## 정답

가능한 한 많은 아이들에게 글러브와 공을 남김없이 나누어 주어야하므로 48과 60의 최대공약수를 구한다. 연산 결과 48과 60의 최대공약수는 12이므로 12개의 상자에 글러브와 공을 각각 4개와 5개씩 담으면 글러브와 공을 하나도 남기지 않고 최대한 많은 학생들한테 나눠줄 수 있다.

|   |    |    |
|---|----|----|
| 2 | 48 | 60 |
| 2 | 24 | 30 |
| 3 | 12 | 15 |
|   | 4  | 5  |


 최대공약수 :  
 $2 \times 2 \times 3 = 12$

# Section 04

## 순서도와 의사코드



## ■ 순서도의 개념

- 다양한 기호를 사용하여 알고리즘을 표현한 것
- 순서도 역시 명확하고 실현 가능한 흐름으로 작성해야 함



그림 2-18 순서도의 역할



## ■ 순서도 기호

표 2-2 순서도 기호

| 기호  | 이름   | 설명   |
|---|------|--|
|    | 흐름선  | 작업의 흐름을 나타낸다.                                  |
|    | 단말   | 순서도의 시작과 끝을 나타낸다.                              |
|    | 준비   | 작업 단계 시작 전 준비를 나타낸다.                           |
|    | 처리   | 처리해야 할 작업을 명시한다.                               |
|    | 판단   | 조건 연산을 나타낸다. 연산 결과(True, False)에 따라 흐름선이 선택된다. |
|  | 입출력  | 데이터의 입력과 출력을 나타낸다.                             |
|  | 서브루틴 | 다른 곳에 정의된 서브프로그램을 호출한다.                        |



## ■ 아르바이트비 지급 순서도

- 출근을 했다면 아르바이트비를 지급하고, 출근하지 않았다면 지급하지 않음

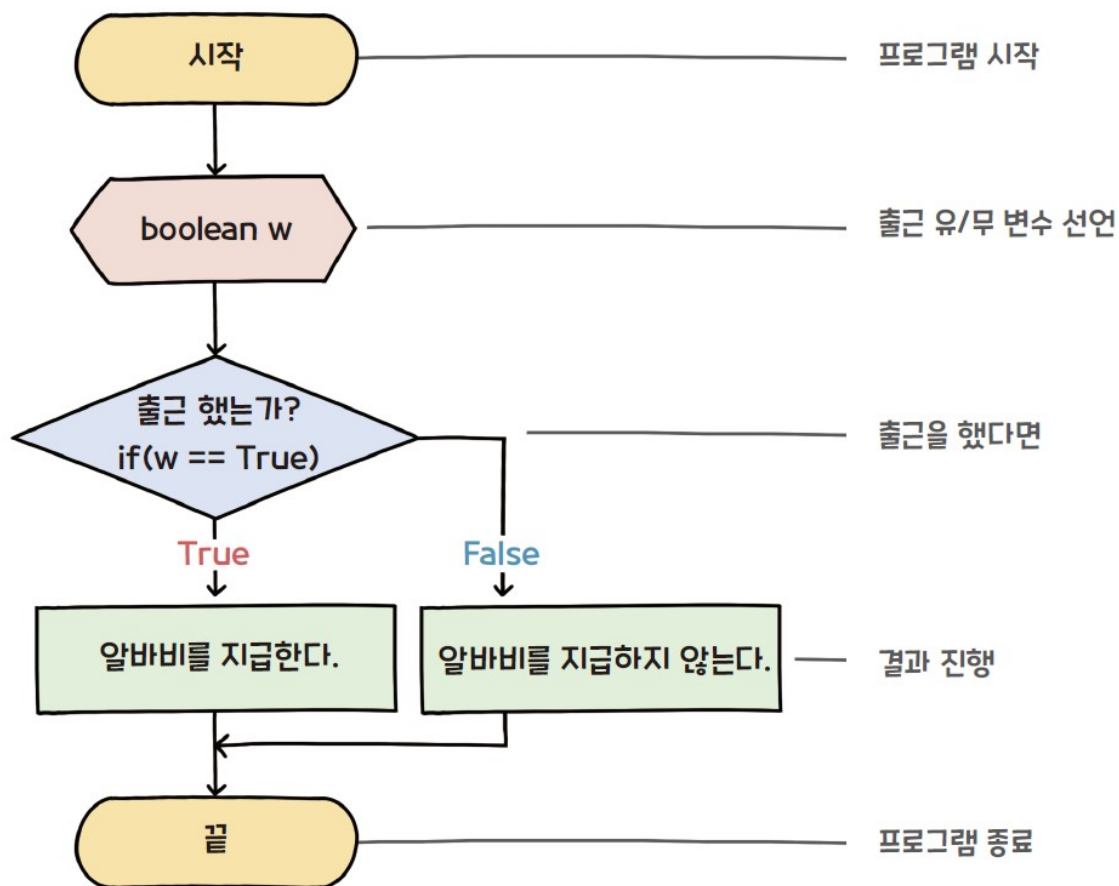


그림 2-19 아르바이트비 지급 순서도



## ■ 홀짝 구분하기

- 입력한 숫자가 홀수인지 짝수인지 구분하는 순서도
- 사용자가 입력한 정수를 2로 나눈 나머지가 '0'이면 '짝수', '1'이면 '홀수'

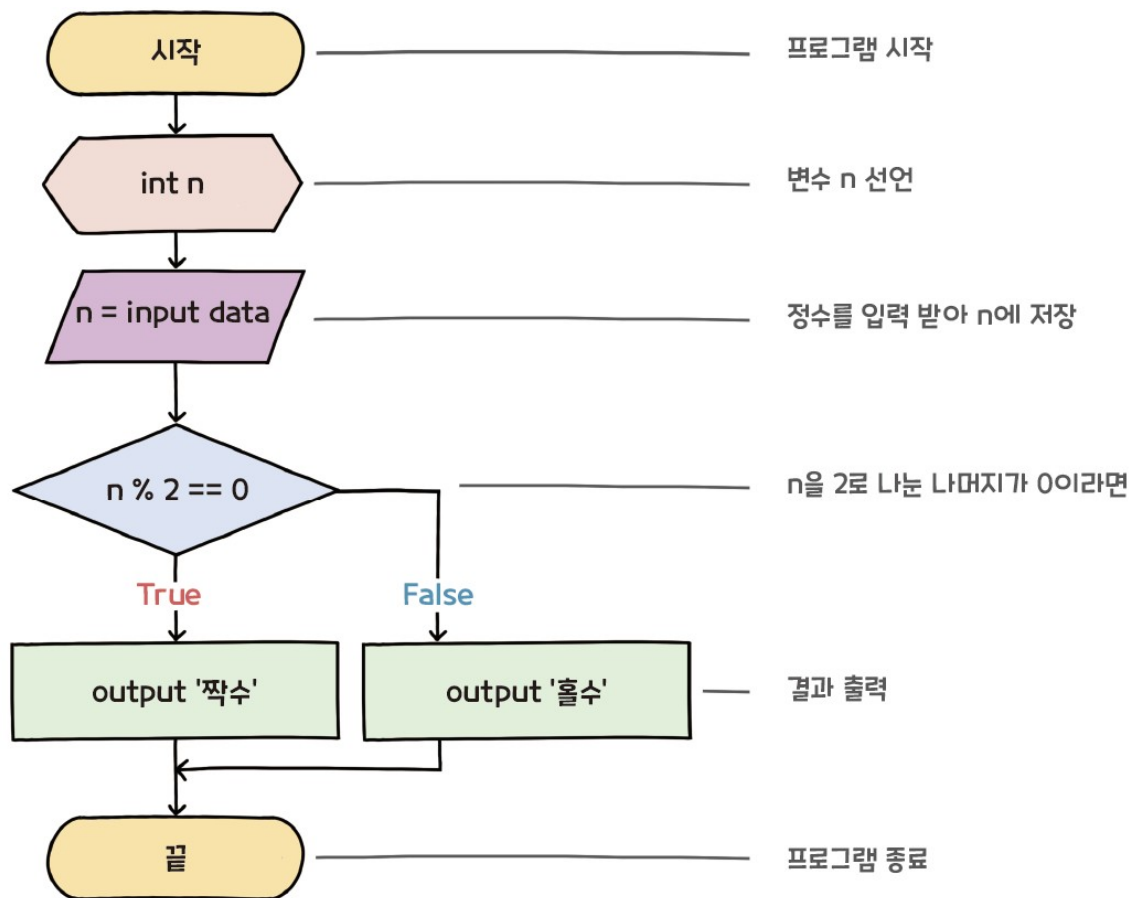


그림 2-20 홀짝 구분하기 순서도



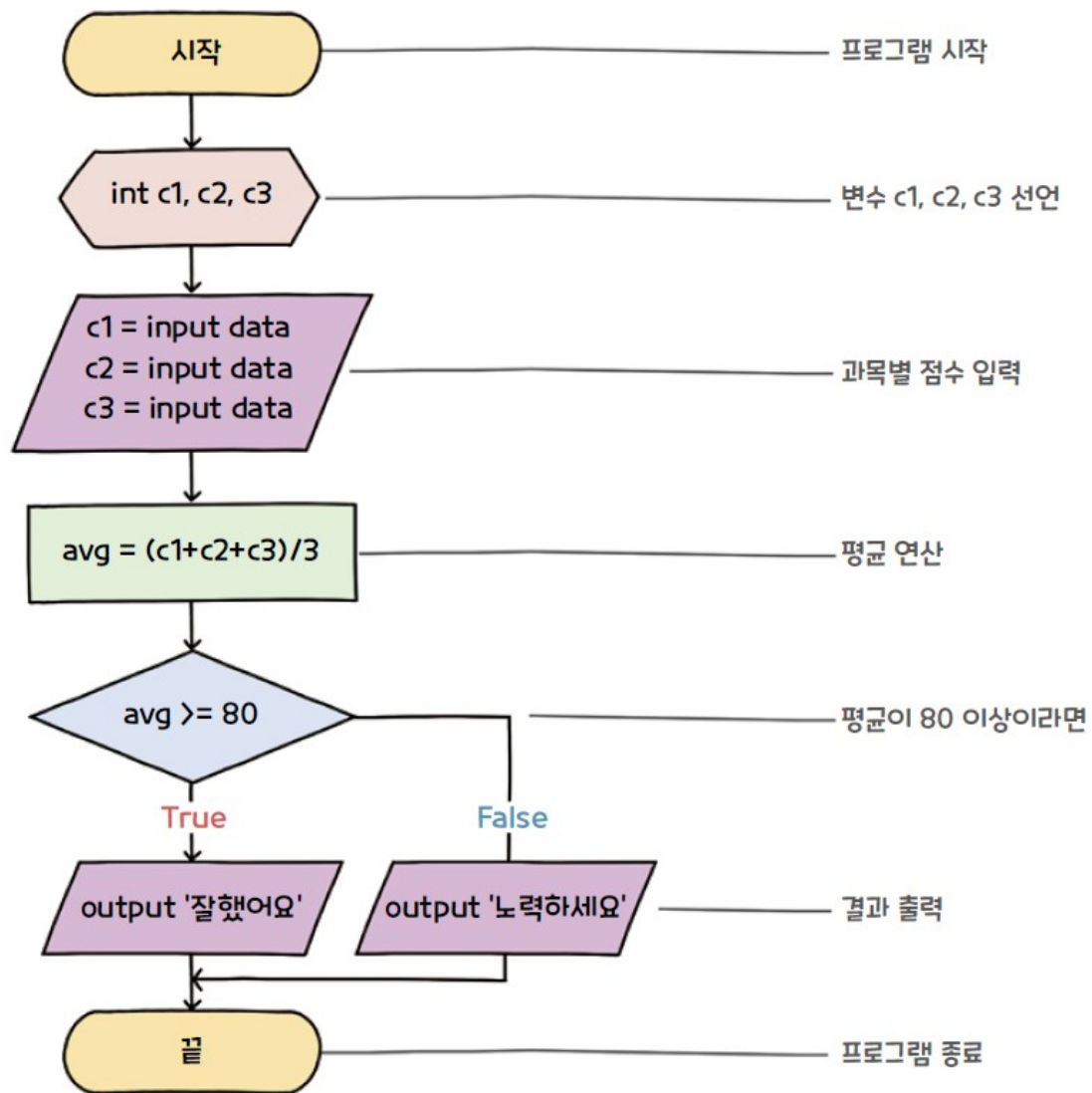
## 확인문제

사용자가 과목별 점수를 입력하면 평균을 계산한 후 80점 이상이면 ‘잘했어요’를, 그렇지 않으면 ‘노력하세요.’를 출력하는 순서도를 작성하시오.





정답





## ■ 의사코드(슈도코드, pseudocode)의 개념

- 순서도와 마찬가지로 알고리즘을 기술하기 위한 방법 중 하나
- 우리가 사용하는 자연어를 이용하여 프로그래밍 언어와 비슷하게 만든 코드이나 실제 컴퓨터에서 실행되지는 않음



## ■ 두 수 교환하기

■ x가 3, y가 5일 때 두 수를 교환하는 알고리즘

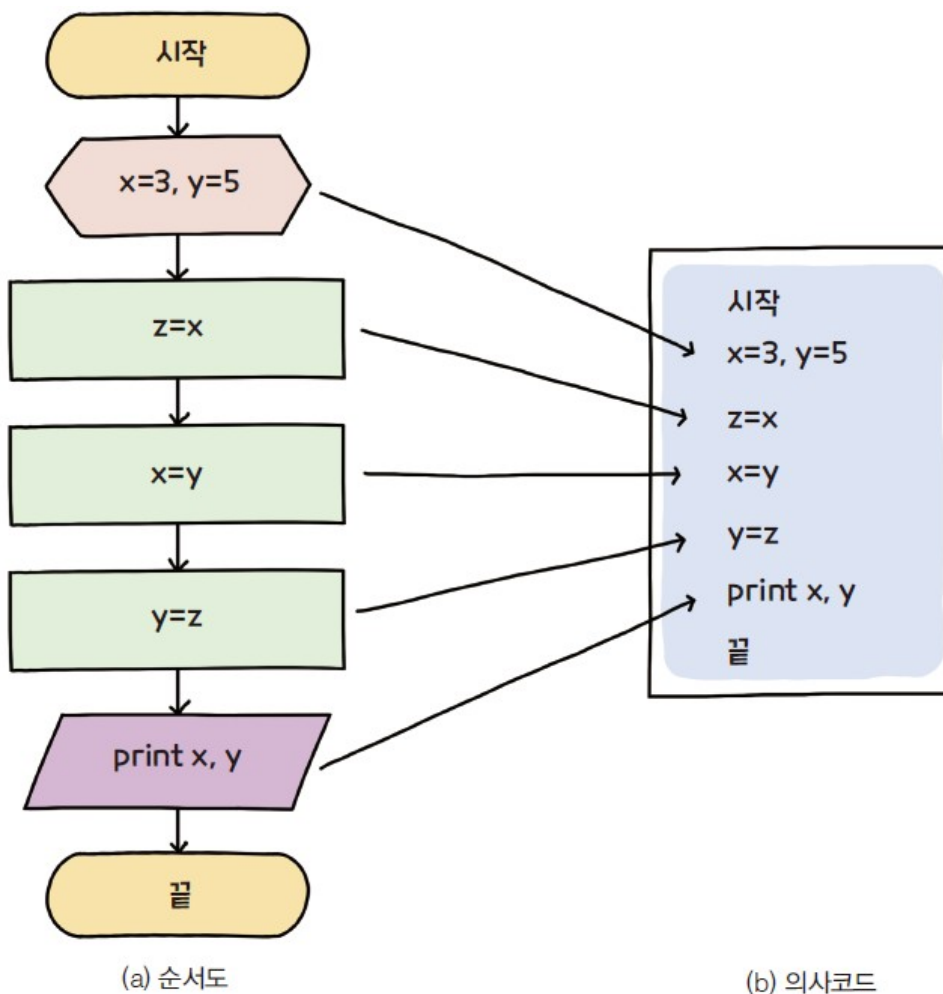


그림 2-21 교환 알고리즘



## ■ 온도 자동 조절하기

- 실내 온도에 따라 ON, OFF를 설정

```
사용자 설정 온도 = input()
에어컨 온도 = current temperature in machine
if(에어컨 온도 > 사용자 설정 온도)
    에어컨 작동 'ON'
else
    에어컨 작동 'OFF'
```

그림 2-22 온도 자동 조절 프로그램



## 확인문제

스캔 받은 아이디 카드를 판별하여 국적에 따른 인사말을 출력하는 의사코드를 작성하시오.

### 정답

```
사용자 정보 = scan ID card
국적 = get nationality from user information

if(국적 == 'Korea')
    print '안녕하세요.'

if(국적 == 'USA')
    print 'Hello'

if(국적 == 'Japan')
    print 'こんにちは'

if(국적 == 'China')
    print '你好'
```

# Thank you!