

# Chapter 14

## 컴퓨팅 사고 프로젝트



# 목차

1. 문자 메시지 요금 계산하기
2. 도시형 텃밭 구역 나누기
3. 전쟁 압호 프로그램
4. 축구 국가대표 선수 연령 파악하기

## Section 01

# 문자 메시지 요금 계산하기



## ■ 문자 메시지 요금 청구 프로그램

- 이동 통신사에서 다음 요금표에 따라 문자 메시지 요금을 계산함

표 14-1 문자 메시지 요금표

구분	회원 등급		
	VVIP	VIP	GOLD
SMS	무제한	100건까지 무제한 101건부터 1건당 10원	10건까지 무제한 11건부터 1건당 10원
MMS	50건까지 무제한 51건부터 1건당 20원	10건까지 무제한 11건부터 1건당 20원	5건까지 무제한 6건부터 1건당 20원



## ■ 알고리즘

- ① 이용자의 회원 등급을 입력한다.
- ② SMS 발송 건수를 입력한다.
- ③ MMS 발송 건수를 입력한다.
- ④ 회원 등급에 따라 요금표를 기준으로 요금을 계산한다.
- ⑤ 청구 요금을 출력한다.



## ■ 순서도로 변환

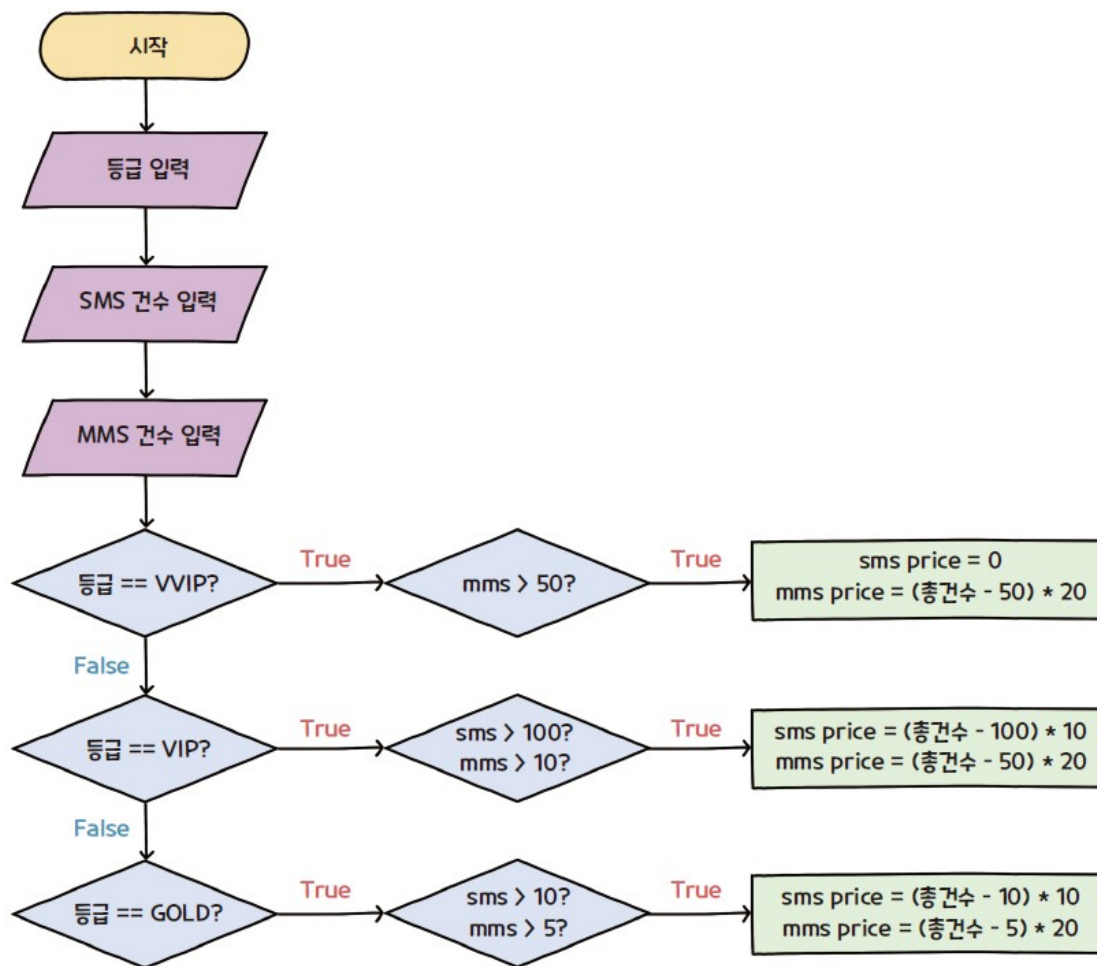


그림 14-1 문자 메시지 요금 계산 순서도



코드 14-1

ch14\_01py

```
01 userGrade = int(input('회원 등급을 선택하세요.\n1. VVIP \t2. VIP \t3. GOLD '))
02 smsCount = int(input('SMS 발송 건수를 입력하세요. '))
03 mmsCount = int(input('MMS 발송 건수를 입력하세요. '))
04 smsPrice = 0
05 mmsPrice = 0
06
07 if userGrade == 1:
08     smsPrice = 0
09
10     if mmsCount > 50:
11         mmsPrice = (mmsCount - 50) * 20
12     else:
13         mmsPrice = 0
14
15 elif userGrade == 2:
16     if smsCount > 100:
17         smsPrice = (smsCount - 100) * 10
18     else:
19         smsPrice = 0
20
21     if mmsCount > 10:
22         mmsPrice = (mmsCount - 10) * 20
23     else:
24         mmsPrice = 0
25
```

```
26 elif userGrade == 3:
27     if smsCount > 10:
28         smsPrice = (smsCount - 10) * 10
29     else:
30         smsPrice = 0
31
32     if mmsCount > 5:
33         mmsPrice = (mmsCount - 5) * 20
34     else:
35         mmsPrice = 0
36
37 print('smsPrice :', smsPrice, '원')
38 print('mmsPrice :', mmsPrice, '원')
```

회원 등급을 선택하세요.

1.VVIP 2.VIP 3.GOLD 1

SMS 발송 건수를 입력하세요. 150

MMS 발송 건수를 입력하세요. 50

smsPrice : 0 원

mmsPrice : 0 원

회원 등급을 선택하세요.

1.VVIP 2.VIP 3.GOLD 3

SMS 발송 건수를 입력하세요. 80

MMS 발송 건수를 입력하세요. 10

smsPrice : 700 원

mmsPrice : 100 원

## Section 02

# 도시형 텃밭 구역 나누기





## ■ 주말농장의 텃밭 구역 나누기

- 텃밭의 전체 길이는 가로 120m, 세로 50m
- 남는 텃밭이 없도록 1인당 최대한 넓은 면적을 분양하려고 할 때 다음 문제를 해결하기
- **문제A** : 텃밭을 정사각형으로 나눈다고 할 때 최대한 넓은 면적을 나누기 위한 가로, 세로의 길이를 구하시오.
- **문제B** : 정사각형으로 나뉜 텃밭의 개수를 구하시오.

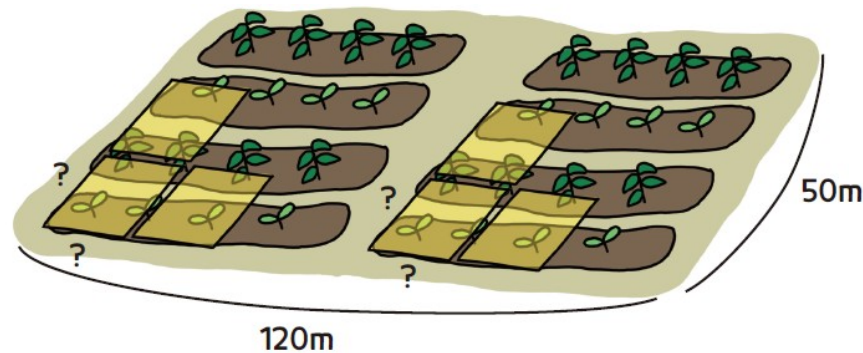


그림 14-2 주말농장의 텃밭 구역 나누기 문제



## ■ 알고리즘-1

- ① 정사각형으로 최대한 넓은 면적을 똑같이 나누기 위해서는 약수를 이용한 최대공약수를 구하면 된다.
- ② 120과 50의 약수를 구한다.
  - 120의 약수 : 1, 2, 3, 4, 5, 6, 8, 10, 12, 15, 20, 24, 30, 40, 60, 120
  - 50의 약수 : 1, 2, 5, 10, 25, 50
- ③ 공통된 약수를 구한다.
  - 공통 약수 : 1, 2, 10
- ④ 최대공약수 : 10
- ⑤ 텃밭 하나의 크기는 가로와 세로가 각각 10m인 정사각형으로 한다.



## ■ 알고리즘 - 2

■ 알고리즘-1을 컴퓨터 프로그램으로 만들기 위해 알고리즘 수정

- ① 전체 가로 길이 120을 1부터 120까지의 정수로 나눠서 나머지가 0인 숫자를 찾는다.
- ② 전체 세로 길이 50을 1부터 50까지의 정수로 나눠서 나머지가 0인 숫자를 찾는다.
- ③ ①과 ②의 숫자 중에서 공통된 숫자를 찾는다.
- ④ ③의 공통된 숫자 중에서 가장 큰 수를 찾는다.



코드 14-2

ch14\_02.py

```
01  wGnd = 120
02  hGnd = 50
03
04  wDivisor = []
05  hDivisor = []
06
07  maxDivisor = 0
08
09  for i in range(1, 120):
10      if wGnd % i == 0:
11          wDivisor.append(i)
12
13  for i in range(1, 51):
14      if hGnd % i == 0:
15          hDivisor.append(i)
16
17  for h in hDivisor:
18      for w in wDivisor:
19          if h == w:
20              maxDivisor = h
21
22  print('최대 가로, 세로 길이 : ', maxDivisor)
```

최대 가로, 세로 길이 : 10



## ■ 알고리즘

■ 전체 가로와 세로 길이를 10m로 나누면 가로와 세로의 개수를 알 수 있으므로, 이를 곱하면 전체 텃밭의 개수를 구할 수 있음

① 전체 가로 길이를 10으로 나눠서 텃밭의 가로 개수를 구한다.

-  $120\text{m} \div 10\text{m} = 12\text{개}$

② 전체 세로 길이를 10으로 나눠서 텃밭의 세로 개수를 구한다.

-  $50\text{m} \div 10\text{m} = 5\text{개}$

③ 가로 개수와 세로 개수를 곱해 전체 텃밭의 개수를 구한다.

-  $12\text{개} \times 5\text{개} = 60\text{개}$



코드 14-3

ch14\_03.py

```
01  ...생략([코드 14-2]의 내용)...  
02  
03  wCnt = wGnd / maxDivisor  
04  hCnt = hGnd / maxDivisor  
05  
06  print('전체 텃밭의 개수 : ', int(wCnt * hCnt))
```

최대 가로, 세로 길이 : 10

전체 텃밭의 개수 : 60

# Section 03

## 전쟁 암호 프로그램



## ■ 전쟁 암호 프로그램

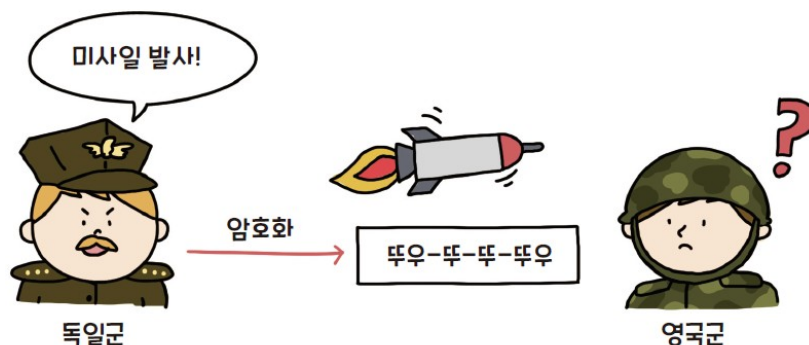


그림 14-3 독일군의 암호 에니그마

### 암호 규칙

- ① 첫 번째와 두 번째 문자 사이에 'a'를 삽입한다.
- ② 세 번째 와 네 번째 문자 사이에 'pp'를 삽입한다.
- ③ 여섯 번째와 일곱 번째 문자 사이에 'le.'를 삽입한다.
- ④ 일곱 번째 문자부터는 과정 ①~③을 반복한다.

- **문제A** : 문자열을 입력하면 암호 규칙에 따라 암호화하는 프로그램을 만드시오.
- **문제B** : 암호화된 문자열을 암호 규칙에 따라 해독하는 프로그램을 만드시오.





## ■ 알고리즘

- 암호 규칙을 보고 'apple.' 문자열이 삽입되는 규칙을 찾아서 알고리즘(해결방법)으로 작성해보기

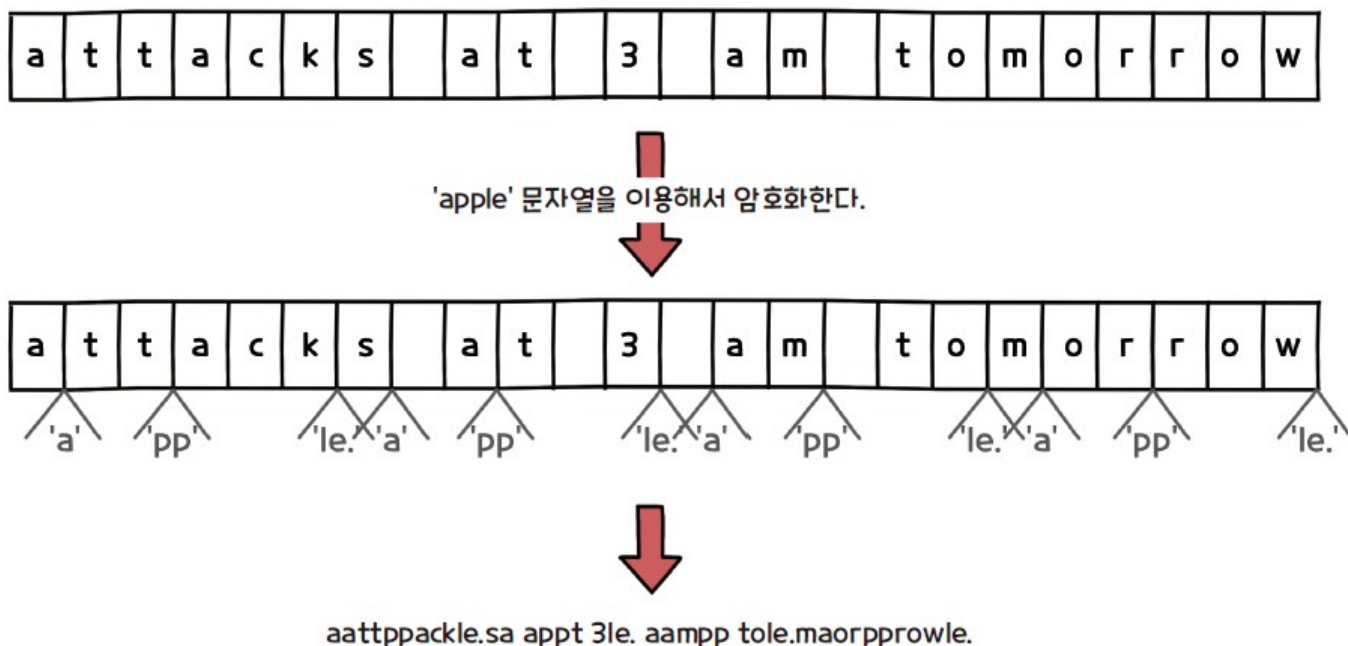


그림 14-4 'apple' 문자열을 이용한 암호화 방법



## ■ 알고리즘

- ① 암호 규칙을 통해서 삽입되는 문자 수만큼 건너뛰면서 문자열이 삽입되는 것을 알 수 있다.
- ② 입력한 문자열을 순회하면서 자리수가 1이면 'a'를 삽입한다.
- ③ 입력한 문자열을 순회하면서 자리수가 3이면 'pp'를 삽입한다.
- ④ 입력한 문자열을 순회하면서 자리수가 6이면 'le.'를 삽입한다.
- ⑤ 입력한 문자열을 순회하면서 자리수가 6보다 크면 자리수를 1로 재설정한다.

# [문제A]의 파이썬 프로그래밍



코드 14-4

ch14\_04.py

```
01  commandStr = input('문자열을 입력하세요.\n')
02  commandStrList = []
03  secretList = ['a', 'pp', 'le.']
04
05  cnt = 0
06  strIndex = 0
07
08  for c in commandStr:
09      commandStrList.append(c)
10
11      strIndex += 1
12      if strIndex > 6: strIndex = 1
13
14      if strIndex == 1:
15          commandStrList.append(secretList[cnt])
16          cnt += 1
17
18      elif strIndex == 3:
19          commandStrList.append(secretList[cnt])
20          cnt += 1
21
22      elif strIndex == 6:
23          commandStrList.append(secretList[2])
24          cnt += 1
25          if cnt >= 3: cnt = 0
26
```

```
27  secretCommandStr = ''
28  for c in commandStrList:
29      secretCommandStr += c
30
31  print(secretCommandStr)
```

문자열을 입력하세요.

attacks at 3 am tomorrow

암호문 : aattppackle.sa appt 3le. aampp tole.maorpprowle.



## ■ 알고리즘

- ① 암호화된 문자열을 리스트에 저장한다.
- ② 리스트의 인덱스가 0인 경우에 해당 문자를 `commandStr`에 추가한다.
- ③ 리스트의 인덱스가 2 또는 3인 경우에 해당 문자를 `commandStr`에 추가한다.
- ④ 리스트의 인덱스가 6 또는 7 또는 8인 경우에 해당 문자를 `commandStr`에 추가한다.
- ⑤ 암호 문자가 제거된 `commandStr`를 출력한다.



코드 14-5

ch14\_05.py

```
01 secretCommandStr = 'aattppackle.sa appt 3le. aampp tole.maorpprowle.'
02 commandStrList = []
03 commandStr = ''
04
05 for c in secretCommandStr:
06     commandStrList.append(c)
07
08 commandIndex = 0
09
10 for i in range(0, len(commandStrList)):
11
12     if commandIndex == 0:
13         commandStr += commandStrList[i]
14     elif commandIndex == 2 or commandIndex == 3:
15         commandStr += commandStrList[i]
16     elif commandIndex == 6 or commandIndex == 7 or commandIndex == 8:
17         commandStr += commandStrList[i]
18
19     commandIndex += 1
20     if(commandIndex >= 12): commandIndex = 0
21
22     print(commandStr)
```

```
a
a
at
att
...생략...
attacks at 3 am tomorrow
```

# Section 04

## 축구 국가대표 선수 연령 파악하기



## ■ 축구 국가대표 선수 11명의 선수 명단을 보고 연령 파악하기



그림 14-5 축구 국가대표 나이

### 선수 명단

홍축구(18세)	남태희(20세)	엄길동(17세)	이강인(18세)	이동준(22세)	황의조(27세)
황의찬(23세)	권창훈(25세)	손준호(27세)	손흥민(27세)	이재성(27세)	

- **문제A** : 18세 이하 선수가 몇 명인지 출력하시오.
- **문제B** : 나이를 내림차순으로 정렬하고 나이가 가장 많은 선수와 가장 적은 선수가 몇 살인지 출력하시오.
- **문제C** : [문제B]를 이용하여 나이가 가장 많은 선수와 가장 적은 선수의 인원수를 각각 출력하시오.



## ■ 알고리즘 - 1

- 기본적인 알고리즘(해결방법)을 생각해 보기

- ① 모든 선수들의 나이를 리스트에 저장한다.
- ② playerUnder18 변수를 선언하고 0으로 초기화한다.
- ③ 반복 실행을 통해서 리스트에 저장된 나이가 18세 이하인지 비교한다.
- ④ 만약 나이가 18이하라면 playerUnder18에 1을 더한다.
- ⑤ 변수 playerUnder18에 저장 값을 출력한다.





## ■ 알고리즘 - 2

■ 알고리즘-1을 프로그램으로 만들기 위해서 알고리즘-1을 프로그래밍 언어와 비슷하게 수정해 보기

- ① 리스트 타입의 playerAgeList에 선수들의 나이를 저장한다.
- ② playerUnder18 변수를 선언하고 0으로 초기화한다.
- ③ 반복문을 이용해서 리스트의 모든 나이를 조회한다.
- ④ 만약 나이가 18보다 작거나 같다면 playerUnder18을 1 증가한다.
- ⑤ playerUnder18의 값을 출력한다.



## ■ 순서도로 변환

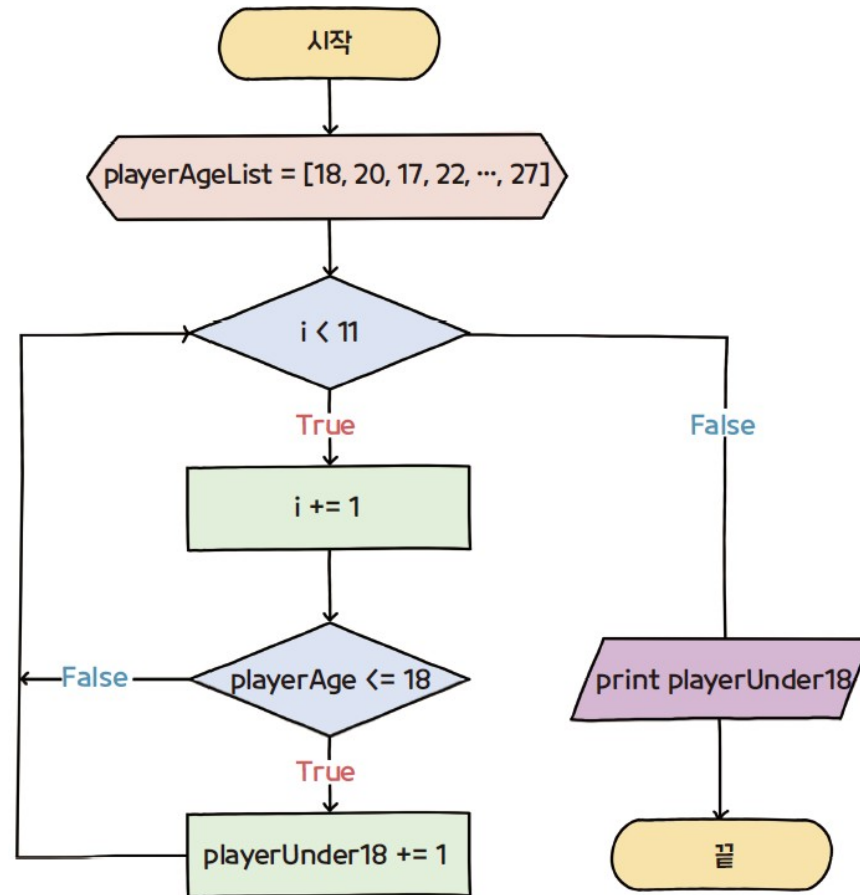


그림 14-6 18세 이하인 선수가 몇 명인지 구하는 순서도



코드 14-6

ch14\_06.py

```
01 playerAgeList = [18, 20, 17, 18, 22, 27, 23, 25, 27, 27]
02 playerUnder18 = 0
03
04 for playerAge in playerAgeList:
05     if playerAge <= 18:
06         playerUnder18 += 1
07
08 print('18세 이하 선수는', playerUnder18, '명입니다.')
```

18세 이하 선수는 3 명입니다.



## ■ 1단계: 문제 분해

- 컴퓨팅 사고의 첫 번째 단계는 문제 분해였음
- [문제B]를 다음과 같이 작은 단위로 분해할 수 있음

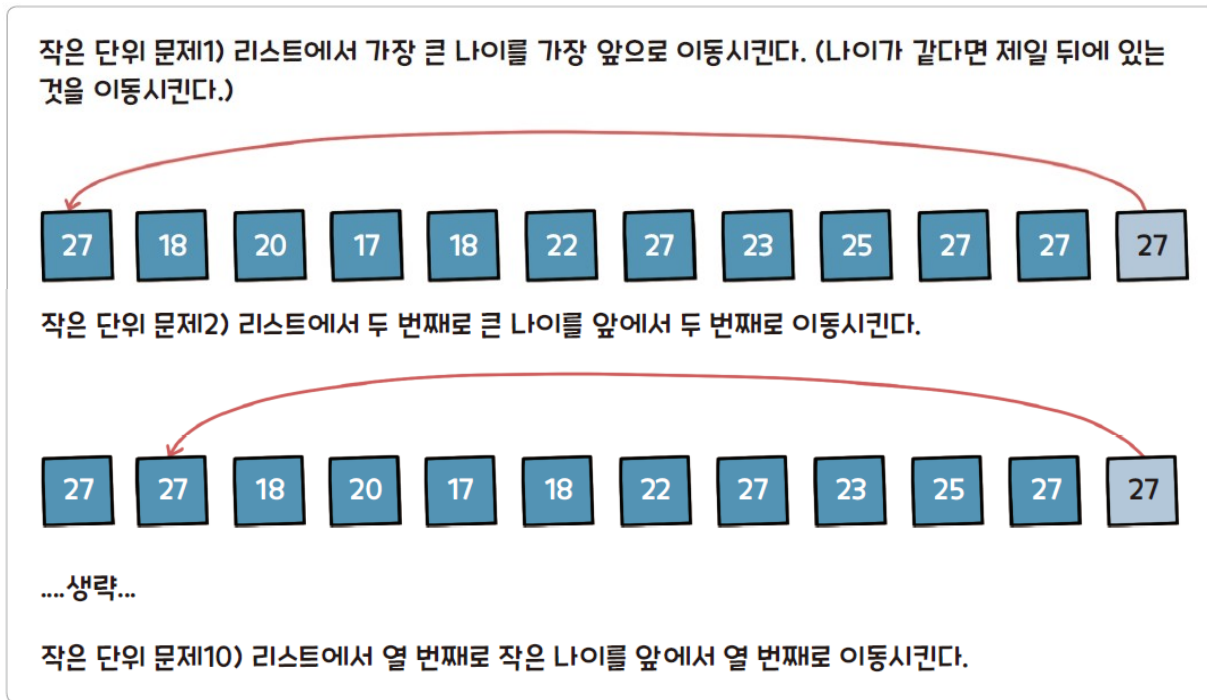


그림 14-7 내림차순을 위한 문제 분해



## ■ 2단계 : 패턴인식

- 작은 단위로 분해한 문제에서 일정한 패턴 찾기
  - 정렬되지 않은 상태에서 각각의 나이는 자신보다 뒤에 있는 숫자들과 비교해서 가장 큰 나이와 자리를 바꿈
- 즉 나이 중 가장 많은 나이의 위치를 가장 앞에 있는 나이의 위치와 바꿈

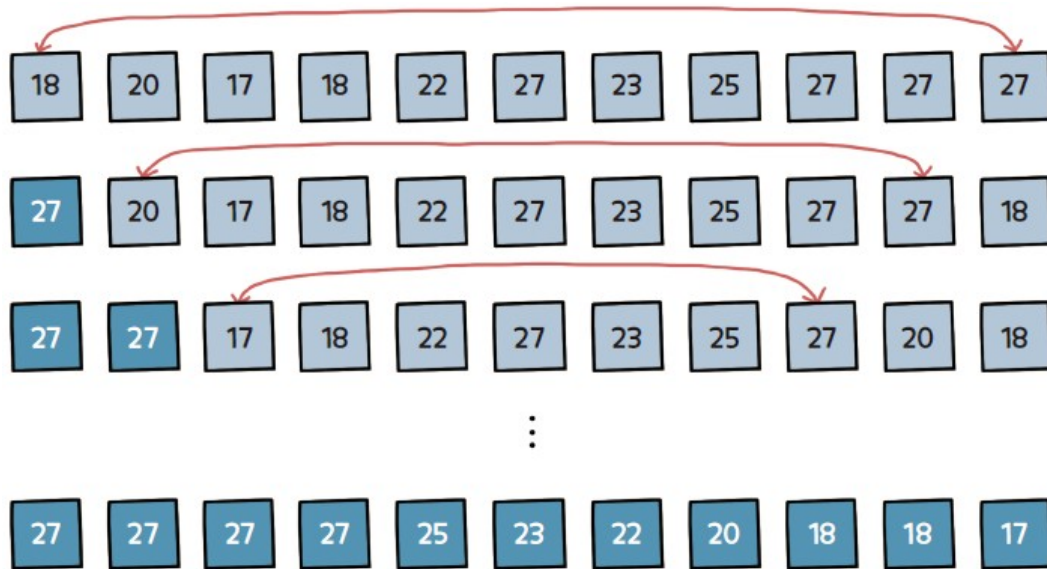


그림 14-8 내림차순 정렬 과정



## ■ 3단계 : 알고리즘

■ 2단계에서 일정한 규칙을 발견했으므로 이제 문제 해결을 위한 알고리즘을 생각해 보기

- ① 리스트에 모든 나이를 저장한다.
- ② 첫 번째 나이부터 마지막 나이까지 반복 실행한다.
- ③ ②의 반복 실행 과정에서 자신보다 뒤에 있는 나이들과 크고 작음을 비교하기 위해서 다시 반복 실행한다.
- ④ ②, ③의 반복 실행 과정에서 자신보다 큰 나이가 있을 경우 자리바꿈을 한다.
- ⑤ 재정렬된 리스트를 출력한다.
- ⑥ 가장 많은 나이를 출력하기 위해서 리스트의 첫 번째 나이를 출력한다.
- ⑦ 가장 적은 나이를 출력하기 위해서 리스트의 마지막 나이를 출력한다.



코드 14-7

ch14\_appEx\_04\_02.py

```
01 playerAgeList = [18, 20, 17, 18, 22, 27, 23, 25, 27, 27, 27]
02
03 for currentIndex in range(0, len(playerAgeList)):
04     max = playerAgeList[currentIndex]
05     maxIndex = currentIndex
06
07     for compareIndex in range(currentIndex, len(playerAgeList)):
08         if playerAgeList[compareIndex] >= max:
09             max = playerAgeList[compareIndex]
10             maxIndex = compareIndex
11
12     max = playerAgeList[maxIndex]
13     playerAgeList[maxIndex] = playerAgeList[currentIndex]
14     playerAgeList[currentIndex] = max
15
16 print(playerAgeList)
17 print('가장 많은 나이 : ', playerAgeList[0])
18 print('가장 적은 나이 : ', playerAgeList[len(playerAgeList) - 1])
```

자신보다 뒤에 있는  
숫자들 중 가장 큰  
숫자와 자리 교환

[27, 27, 27, 27, 25, 23, 22, 20, 18, 18, 17]

가장 많은 나이 : 27

가장 적은 나이 : 17



## ■ 알고리즘

- [문제B]에서 선수들의 나이를 내림차순으로 정렬했기 때문에 가장 첫 번째 나이가 가장 큰 나이고, 가장 마지막 나이가 가장 적은 나이임
  - 따라서 나이 리스트를 반복 실행하면서 가장 큰 나이와 가장 작은 나이의 개수를 구하면 됨
- ① 첫 번째 나이를 maxAge에 저장한다.
  - ② 마지막 나이를 minAge에 저장한다.
  - ③ maxCount, minCount를 선언하고 0으로 초기화한다.
  - ④ maxCount, minCount를 선언하고 0으로 초기화한다.
  - ⑤ 리스트를 반복하면서 maxAge와 minAge가 몇 개인지 카운트하기 위해 age가 maxAge와 같다면 maxCount를 1 증가시키고, age가 minAge와 같다면 minCount를 1 증가시킨다.
  - ⑥ maxCount와 minCount를 출력한다





코드 14-8

ch14\_08.py

```
01  ...생략([코드 14-7]의 내용)...
02
03  maxAge = playerAgeList[0]
04  minAge = playerAgeList[len(playerAgeList) - 1]
05  maxCount = 0
06  minCount = 0
07
08  for age in playerAgeList:
09      if age == maxAge:
10          maxCount += 1
11      if age == minAge:
12          minCount += 1
13
14  print('가장 많은 나이의 개수 : ', maxCount)
15  print('가장 적은 나이의 개수 : ', minCount)
```

[27, 27, 27, 27, 25, 23, 22, 20, 18, 18, 17]

가장 많은 나이 : 27

가장 적은 나이 : 17

가장 많은 나이의 개수 : 4

가장 적은 나이의 개수 : 1

# Thank you!