

# 파이썬 문법 익히기

## 연습문제

## 7. 파이썬 분기문

101.

▶ 파이썬에서 True 혹은 False를 갖는 데이터 타입은 무엇인가?

102.

▶ 아래 코드의 출력 결과를 예상하라

```
print(3 == 5)
```

103.

▶ 아래 코드의 출력 결과를 예상하라

```
print(3 < 5)
```

104.

▶ 아래 코드의 결과를 예상하라.

```
x = 4  
print(1 < x < 5)
```

105.

▶ 아래 코드의 결과를 예상하라.

```
print ((3 == 3) and (4 != 3))
```

**106.**

▶ 아래 코드에서 에러가 발생하는 원인에 대해 설명하라.

```
print(3 => 4)
```

**107.**

▶ 아래 코드의 출력 결과를 예상하라

```
if 4 < 3:  
    print("Hello World")
```

**108.**

▶ 아래 코드의 출력 결과를 예상하라

```
if 4 < 3:  
    print("Hello World.")  
else:  
    print("Hi, there.")
```

**109.**

▶ 아래 코드의 출력 결과를 예상하라

```
if True :  
    print ("1")  
    print ("2")  
else :  
    print("3")  
print("4")
```

## 110.

- ▶ 아래 코드의 출력 결과를 예상하라

```
if True :  
    if False:  
        print("1")  
        print("2")  
    else:  
        print("3")  
else :  
    print("4")  
print("5")
```

## 111.

- ▶ 사용자로부터 입력받은 문자열을 두 번 출력하라. 아래는 사용자가 "안녕하세요"를 입력한 경우의 출력 결과이다.

```
>> 안녕하세요  
안녕하세요안녕하세요
```

## 112.

- ▶ 사용자로부터 하나의 숫자를 입력받고, 입력 받은 숫자에 10을 더해 출력하라.

```
>> 숫자를 입력하세요: 30  
40
```

## 113.

- ▶ 사용자로부터 하나의 숫자를 입력 받고 짝수/홀수를 판별하라.

```
>> 30  
짝수
```

## 114.

- ▶ 사용자로부터 값을 입력받은 후 해당 값에 20을 더한 값을 출력하라. 단 사용자가 입력한 값과 20을 더한 계산 값이 255를 초과하는 경우 255를 출력해야 한다.

>> 입력값: 200

출력값: 220

>> 입력값: 240

출력값: 255

## 115.

- ▶ 사용자로부터 하나의 값을 입력받은 후 해당 값에 20을 뺀 값을 출력하라. 단 출력 값의 범위는 0~255이다. 예를 들어 곱빼값이 0보다 작은 값이되는 경우 0을 출력하고 255보다 큰 값이 되는 경우 255를 출력해야 한다.

>> 입력값: 200

출력값: 180

>> 입력값: 15

출력값: 0

## 116.

- ▶ 사용자로부터 입력 받은 시간이 정각인지 판별하라.

>> 현재시간:02:00

정각 입니다.

>> 현재시간:03:10

정각이 아닙니다

## 117.

- ▶ 사용자로 입력받은 단어가 아래 fruit 리스트에 포함되어 있는지를 확인하라. 포함되었다면 "정답입니다"를 아닐 경우 "오답입니다" 출력하라.

```
fruit = ["사과", "포도", "홍시"]
>> 좋아하는 과일은? 사과
정답입니다.
```

## 118.

- ▶ 아래와 같이 입사하고 싶은 IT 기업 리스트가 있을 때, 화면에서 입력한 회사가 입사하고 싶은 회사면 '입사 희망 회사 입니다'라고 출력하고, 그렇지 않으면, '입사 희망 회사가 아닙니다'라고 출력하는 코드를 작성하라.

```
it_company_list = ["Microsoft", "Google", "Naver", "Kakao", "SAMSUNG", "LG"]
```

## 119.

- ▶ 아래와 같이 fruit 딕셔너리가 정의되어 있다. 사용자가 입력한 값이 딕셔너리 키 (key) 값에 포함되었다면 "정답입니다"를 아닐 경우 "오답입니다" 출력하라.

```
fruit = {"봄" : "딸기", "여름" : "토마토", "가을" : "사과"}
>> 제가좋아하는계절은: 봄
정답입니다.
```

## 120.

- ▶ 아래와 같이 fruit 딕셔너리가 정의되어 있다. 사용자가 입력한 값이 딕셔너리 값 (value)에 포함되었다면 "정답입니다"를 아닐 경우 "오답입니다" 출력하라.

```
fruit = {"봄" : "딸기", "여름" : "토마토", "가을" : "사과"}
>> 좋아하는과일은? 한라봉
오답입니다.
```

## 121.

- ▶ 사용자로부터 문자 한 개를 입력 받고, 소문자일 경우 대문자로, 대문자 일 경우, 소문자

로 변경해서 출력하라.

```
>> a
A
```

힌트-1 : `islower()` 함수는 문자의 소문자 여부를 판별합니다. 만약 소문자일 경우 `True`, 대문자일 경우 `False`를 반환합니다.

힌트-2 : `upper()` 함수는 대문자로, `lower()` 함수는 소문자로 변경합니다.

## 122.

▶ 점수 구간에 해당하는 학점이 아래와 같이 정의되어 있다. 사용자로부터 `score`를 입력받아 학점을 출력하라.

점수	학점
81~100	A
61~80	B
41~60	C
21~40	D
0~20	E

```
>> score: 83
grade is A
```

## 123.

▶ 사용자로부터 달러, 엔, 유로, 또는 위안 금액을 입력받은 후 이를 원으로 변환하는 프로그램을 작성하라. 각 통화별 환율은 다음과 같다. 사용자는 100 달러, 1000 엔, 13 유로, 100 위안과 같이 금액과 통화명 사이에 공백을 넣어 입력한다고 가정한다.

통화명	환율
달러	1167
엔	1.096
유로	1268
위안	171

```
>> 입력: 100 달러
116700.00 원
```

## 124.

- ▶ 사용자로부터 세 개의 숫자를 입력 받은 후 가장 큰 숫자를 if 문을 사용하여 출력하라.

```
>> input number1: 10
>> input number2: 9
>> input number3: 20
20
```

## 125.

- ▶ 휴대폰 번호 앞자리에 따라 통신사는 아래와 같이 구분된다. 사용자로부터 휴대전화 번호를 입력 받고, 통신사를 출력하는 프로그램을 작성하라.

```
번호   통신사
011    SKT
016    KT
019    LGU
010    알수없음
>> 휴대전화 번호 입력: 011-345-1922
당신은 SKT 사용자입니다.
```

## 126.

- ▶ 우편번호는 5자리로 구성되는데, 앞의 세자리는 구를 나타낸다. 예를들어, 강북구의 경우 010, 011, 012 세 자리로 시작한다.

-	0	1	2	3	4	5	6	7	8	9
01	강북구	강북구	강북구	도봉구	도봉구	도봉구	노원구	노원구	노원구	노원구

사용자로 부터 5자리 우편번호를 입력받고 구를 판별하라

```
>> 우편번호: 01400
도봉구
```



## 127.

- ▶ 주민등록번호 뒷 자리 7자리 중 첫째 자리는 성별을 나타내는데, 1, 3은 남자 2, 4는 여자를 의미한다. 사용자로부터 13자리의 주민등록번호를 입력 받은 후 성별 (남자, 여자)를 출력하는 프로그램을 작성하라.

>> 주민등록번호: 821010-1635210

남자

## 128.

- ▶ 주민등록번호의 뒷 자리 7자리 중 두번째와 세번째는 지역코드를 의미한다. 주민 등록 번호를 입력 받은 후 출생지가 서울인지 아닌지 판단하는 코드를 작성하라

지역코드      출생지

00 ~ 08      서울

09 ~ 12      부산

>> 주민등록번호: 821010-1635210

서울이 아닙니다.

>> 주민등록번호: 861010-1015210

서울 입니다.

## 129.

- ▶ 주민등록번호는 13자리로 구성되는데 마지막 자리수는 주민등록번호의 유효성을 체크하는데 사용된다. 먼저 앞에서부터 12자리의 숫자에 2, 3, 4, 5, 6, 7, 8, 9, 2, 3, 4, 5를 차례로 곱한 뒤 그 값을 전부 더한다. 연산 결과 값을 11로 나누면 나머지가 나오는데 11에서 나머지를 뺀 값이 주민등록번호의 마지막 번호가 된다.

8 2 1 0 1 0 - 1 6 3 5 2 1 0

x 2 3 4 5 6 7    8 9 2 3 4 5

-----

1차 계산:  $(8*2 + 2*3 + 1*4 + 0*5 + 1*6 + 0*7 + 1*8 + 6*9 + 3*2 + 5*3 + 2*4 + 1*5) = (128 \% 11) = 7$

2차 계산:  $11 - 7 = 4$

위와 같이 821010-1635210에 대해서 계산을 해보면 마지막 자리는 4가 되어야 함을 알 수 있다. 즉, 821010-1635210은 유효하지 않은 주민등록번호임을 알 수 있다.

다음과 같이 사용자로부터 주민등록번호를 입력받은 후 주민등록번호가 유효한지를 출력하는 프로그램을 작성하라.

>> 주민등록번호: 821010-1635210  
유효하지 않은 주민등록번호입니다.

## 130.

▶ 아래 코드는 비트코인의 가격 정보를 딕셔너리로 가져오는 코드이다.

```
import requests
btc = requests.get("https://api.bithumb.com/public/ticker/").json()['data']
btc 딕셔너리 안에는 시가, 종가, 최고가, 최저가 등이 저장되어 있다. 최고가와 최저가의 차이를 변동폭으로 정의할 때 (시가 + 변동폭)이 최고가 보다 높을 경우 "상승장", 그렇지 않은 경우 "하락장" 문자열을 출력하라.
```

Key Name	Description
opening_price	최근 24시간 내 시작 거래금액
closing_price	최근 24시간 내 마지막 거래금액
min_price	최근 24시간 내 최저 거래금액
max_price	최근 24시간 내 최고 거래금액

## 8. 파이썬 반복문

131.

▶ 아래의 과일 리스트를 `for` 문을 사용하여, 한줄 씩 출력하라.

```
fruits = ["사과", "귤", "수박"]
```

132.

▶ `for`문의 실행결과를 예측하라.

```
과일 = ["사과", "귤", "수박"]  
for 변수 in 과일:  
    print("#####")
```

133.

▶ 다음 `for` 문과 동일한 기능을 수행하는 코드를 변수 할당과 `print`문을 사용하여 작성하세요.

```
for 변수 in ["A", "B", "C"]:  
    print(변수)
```

134.

▶ `for`문을 풀어서 동일한 동작을하는 코드를 변수 할당과 `print`문을 사용하여 작성하라.

```
for 변수 in ["A", "B", "C"]:  
    print("출력:", 변수)
```

135.

▶ for문을 풀어서 동일한 동작을 하는 코드를 작성하라.

```
for 변수 in ["A", "B", "C"]:  
    b = 변수.lower()  
    print("변환:", b)
```

**136.**

▶ 다음 코드를 for문으로 작성하라.

```
변수 = 10  
print(변수)  
변수 = 20  
print(변수)  
변수 = 30  
print(변수)
```

**137.**

▶ 다음 코드를 for문으로 작성하라.

```
print(10)  
print(20)  
print(30)
```

**138.**

▶ 다음 코드를 for문으로 작성하라.

```
print(10)  
print("-----")  
print(20)  
print("-----")  
print(30)
```

```
print("-----")
```

**139.**

▶ 다음 코드를 for문으로 작성하라.

```
print("++++")
print(10)
print(20)
print(30)
```

**140.**

▶ 다음 코드를 for문으로 작성하라.

```
print("-----")
print("-----")
print("-----")
print("-----")
```

**141.**

▶ 다음과 같이 판매가가 저장된 리스트가 있을 때 부가세가 포함된 가격을 for 문을 사용해서 화면에 출력하라. 단 부가세는 10원으로 가정한다.

```
리스트 = [100, 200, 300]
110
210
310
```

**142.**

▶ for 문을 사용해서 리스트에 저장된 값을 다음과 같이 출력하라.

리스트 = ["김밥", "라면", "튀김"]

오늘의 메뉴: 김밥

오늘의 메뉴: 라면

오늘의 메뉴: 튀김

## 143.

▶ 리스트에 주식 종목이름이 저장돼 있다.

리스트 = ["SK하이닉스", "삼성전자", "LG전자"]

저장된 문자열의 길이를 다음과 같이 출력하라.

6

4

4

## 144.

▶ 리스트에는 동물이름이 문자열로 저장돼 있다.

리스트 = ['dog', 'cat', 'parrot']

동물 이름과 글자수를 다음과 같이 출력하라.

dog 3

cat 3

parrot 6

## 145.

▶ 리스트에 동물 이름 저장돼 있다.

리스트 = ['dog', 'cat', 'parrot']

for문을 사용해서 동물 이름의 첫 글자만 출력하라.

d

c

p

## 146.

▶ 리스트에는 세 개의 숫자가 할당돼 있다.

리스트 = [1, 2, 3]

for문을 사용해서 다음과 같이 출력하라.

3 x 1

3 x 2

3 x 3

## 147.

▶ 리스트에는 세 개의 숫자가 할당돼 있다.

리스트 = [1, 2, 3]

for문을 사용해서 다음과 같이 출력하라.

3 x 1 = 3

3 x 2 = 6

3 x 3 = 9

## 148.

▶ 리스트에는 네 개의 문자열이 할당돼 있다.

리스트 = ["가", "나", "다", "라"]

for문을 사용해서 다음과 같이 출력하라.

나

다

라

**149.**

▶ 리스트에는 네 개의 문자열이 할당돼 있다.

리스트 = ["가", "나", "다", "라"]

for문을 사용해서 다음과 같이 출력하라.

가

다

**150.**

▶ 리스트에는 네 개의 문자열이 할당돼 있다.

리스트 = ["가", "나", "다", "라"]

for문을 사용해서 다음과 같이 출력하라.

라

다

나

가

**151.**

▶ 리스트에는 네 개의 정수가 저장돼 있다.

리스트 = [3, -20, -3, 44]

for문을 사용해서 리스트의 음수를 출력하라.

-20

-3

**152.**



▶ for문을 사용해서 3의 배수만을 출력하라.

```
리스트 = [3, 100, 23, 44]
```

```
3
```

**153.**

▶ 리스트에서 20 보다 작은 3의 배수를 출력하라

```
리스트 = [13, 21, 12, 14, 30, 18]
```

```
12
```

```
18
```

**154.**

▶ 리스트에서 세 글자 이상의 문자를 화면에 출력하라

```
리스트 = ["I", "study", "python", "language", "!"]
```

```
study
```

```
python
```

```
language
```

**155.**

▶ 리스트에서 대문자만 화면에 출력하라.

```
리스트 = ["A", "b", "c", "D"]
```

```
A
```

```
D
```

(참고) `isupper()` 메서드는 대문자 여부를 판별합니다.

```
>> 변수 = "A"
```

```
>> 변수.isupper()
```

```
True
```

```
>> 변수 = "a"
>> 변수.isupper()
False
```

## 156.

▶ 리스트에서 소문자만 화면에 출력하라.

```
리스트 = ["A", "b", "c", "D"]
b
c
```

## 157.

▶ 이름의 첫 글자를 대문자로 변경해서 출력하라.

```
리스트 = ['dog', 'cat', 'parrot']
Dog
Cat
Parrot
(참고) upper() 메서드는 문자열을 대문자로 변경합니다.
```

```
>> 변수 = "a"
>> a.upper()
A
>> 변수 = "abc"
>> 변수.upper()
ABC
```

## 158.

▶ 파일 이름이 저장된 리스트에서 확장자를 제거하고 파일 이름만 화면에 출력하라. (힌트: split() 메서드)

```
리스트 = ['hello.py', 'ex01.py', 'intro.hwp']
```

```
hello
ex01
intro
```

## 159.

- ▶ 파일 이름이 저장된 리스트에서 확장자가 .h인 파일 이름을 출력하라.

```
리스트 = ['intra.h', 'intra.c', 'define.h', 'run.py']
intra.h
define.h
```

## 160.

- ▶ 파일 이름이 저장된 리스트에서 확장자가 .h나 .c인 파일을 화면에 출력하라.

```
리스트 = ['intra.h', 'intra.c', 'define.h', 'run.py']
intra.h
intra.c
define.h
```

## 161.

- ▶ for문과 range 구문을 사용해서 0~99까지 한 라인에 하나씩 순차적으로 출력하는 프로그램을 작성하라.

## 162.

- ▶ 월드컵은 4년에 한 번 개최된다. range()를 사용하여 2002~2050년까지 중 월드컵이 개최되는 연도를 출력하라.

```
2002
2006
2010
```

...

2042

2046

2050

참고) range의 세번째 파라미터는 증감폭을 결정합니다.

```
>> print(list(range(0, 10, 2)))
```

```
[0, 2, 4, 6, 8]
```

## 163.

▶ 1부터 30까지의 숫자 중 3의 배수를 출력하라.

3

6

9

12

15

18

21

24

27

30

## 164.

▶ 99부터 0까지 1씩 감소하는 숫자들을, 한 라인에 하나씩 출력하라.

## 165.

for문을 사용해서 아래와 같이 출력하라.

0.0

0.1

0.2

0.3  
0.4  
0.5  
...  
0.9

## 166.

▶ 구구단 3단을 출력하라.

3x1 = 3  
3x2 = 6  
3x3 = 9  
3x4 = 12  
3x5 = 15  
3x6 = 18  
3x7 = 21  
3x8 = 24  
3x9 = 27

## 167.

▶ 구구단 3단을 출력하라. 단 홀수 번째만 출력한다.

3x1 = 3  
3x3 = 9  
3x5 = 15  
3x7 = 21  
3x9 = 27

## 168.

▶ 1~10까지의 숫자에 대해 모두 더한 값을 출력하는 프로그램을 **for** 문을 사용하여 작성하라.

합 : 55

## 169.

▶ 1~10까지의 숫자 중 모든 홀수의 합을 출력하는 프로그램을 for 문을 사용하여 작성하라.

합: 25

## 170.

▶ 1~10까지의 숫자를 모두 곱한 값을 출력하는 프로그램을 for 문을 사용하여 작성하라.

## 171.

▶ 아래와 같이 리스트의 데이터를 출력하라. 단, for문과 range문을 사용하라.

```
price_list = [32100, 32150, 32000, 32500]
```

```
32100
```

```
32150
```

```
32000
```

```
32500
```

## 172.

▶ 아래와 같이 리스트의 데이터를 출력하라. 단, for문과 range문을 사용하라.

```
price_list = [32100, 32150, 32000, 32500]
```

```
0 32100
```

```
1 32150
```

```
2 32000
```

```
3 32500
```

### 173.

▶ 아래와 같이 리스트의 데이터를 출력하라. 단, for문과 range문을 사용하라.

```
price_list = [32100, 32150, 32000, 32500]
3 32100
2 32150
1 32000
0 32500
```

### 174.

▶ 아래와 같이 리스트의 데이터를 출력하라. 단, for문과 range문을 사용하라.

```
price_list = [32100, 32150, 32000, 32500]
100 32150
110 32000
120 32500
```

### 175.

my\_list를 아래와 같이 출력하라.

```
my_list = ["가", "나", "다", "라"]
가 나
나 다
다 라
```

### 176.

▶ 리스트를 아래와 같이 출력하라.

```
my_list = ["가", "나", "다", "라", "마"]
가 나 다
나 다 라
```

다 라 마

## 177.

▶ 반복문과 range 함수를 사용해서 my\_list를 아래와 같이 출력하라.

```
my_list = ["가", "나", "다", "라"]
```

라 다

다 나

나 가

## 178.

▶ 리스트에는 네 개의 정수가 저장되어 있다. 각각의 데이터에 대해서 자신과 우측값과의 차분값을 화면에 출력하라.

```
my_list = [100, 200, 400, 800]
```

예를들어 100을 기준으로 우측에 위치한 200과의 차분 값을 화면에 출력하고, 200을 기준으로 우측에 위치한 400과의 차분값을 화면에 출력한다. 이어서 400을 기준으로 우측에 위치한 800과의 차분값을 화면에 출력한다.

100

200

400

## 179.

▶ 리스트에는 6일 간의 종가 데이터가 저장되어 있다. 종가 데이터의 3일 이동 평균을 계산하고 이를 화면에 출력하라.

```
my_list = [100, 200, 400, 800, 1000, 1300]
```

첫 번째 줄에는 100, 200, 400의 평균값이 출력된다. 두 번째 줄에는 200, 400, 800의 평균값이 출력된다. 같은 방식으로 나머지 데이터의 평균을 출력한다.

233.33333333333334



466.6666666666667  
733.3333333333334  
1033.3333333333333

## 180.

- ▶ 리스트에 5일간의 저가, 고가 정보가 저장돼 있다. 고가와 저가의 차를 변동폭이라고 정의할 때, low, high 두 개의 리스트를 사용해서 5일간의 변동폭을 volatility 리스트에 저장하라.

```
low_prices = [100, 200, 400, 800, 1000]
high_prices = [150, 300, 430, 880, 1000]
```

## 181.

- ▶ 아래 표에서 하나의 행을 하나의 리스트로, 총 3개의 리스트를 갖는 이차원 리스트 apart를 정의하라.

101호	102호
201호	202호
301호	302호

## 182.

- ▶ 아래 표에서 하나의 열을 하나의 리스트로, 총 2개의 리스트를 갖는 이차원 리스트 stock을 정의하라.

시가	종가
100	80
200	210
300	330

### 183.

- ▶ 아래 표를 **stock** 이름의 딕셔너리로 표현하라. 시가를 **key**로 저장하고, 나머지 같은 열의 데이터를 리스트로 저장해서 **value**로 저장한다. 종가 역시 **key**로 저장하고 나머지 같은 열의 데이터를 리스트로 저장해서 **value**로 저장한다.

시가	종가
100	80
200	210
300	330

### 184.

- ▶ 아래 표를 **stock** 이라는 이름의 딕셔너리로 표현하라. 날짜를 **key**로 저장하고, 나머지 같은 행의 데이터를 리스트로 저장해서 **value**로 저장한다. 첫 열이 날짜이다.

10/10	80	110	70	90
10/11	210	230	190	200

### 185.

리스트에 저장된 데이터를 아래와 같이 출력하라.

```
apart = [ [101, 102], [201, 202], [301, 302] ]
```

101 호

102 호

201 호

202 호

301 호

302 호

### 186.

- ▶ 리스트에 저장된 데이터를 아래와 같이 출력하라.

```
apart = [ [101, 102], [201, 202], [301, 302] ]
```

301 호

302 호

201 호

202 호

101 호

102 호

**187.**

▶ 리스트에 저장된 데이터를 아래와 같이 출력하라.

```
apart = [ [101, 102], [201, 202], [301, 302] ]
```

302 호

301 호

202 호

201 호

102 호

101 호

**188.**

▶ 리스트에 저장된 데이터를 아래와 같이 출력하라.

```
apart = [ [101, 102], [201, 202], [301, 302] ]
```

101 호

-----

102 호

-----

201 호

-----

202 호

-----

301 호

-----

302 호

-----

## 189.

▶ 리스트에 저장된 데이터를 아래와 같이 출력하라.

```
apart = [ [101, 102], [201, 202], [301, 302] ]  
101 호  
102 호  
-----  
201 호  
202 호  
-----  
301 호  
302 호  
-----
```

## 190.

▶ 리스트에 저장된 데이터를 아래와 같이 출력하라.

```
apart = [ [101, 102], [201, 202], [301, 302] ]  
101 호  
102 호  
201 호  
202 호  
301 호  
302 호  
-----
```

## 191.

▶ data에는 매수한 종목들의 OHLC (open/high/low/close) 가격 정보가 할당 되어있다.

```
data = [  
    [ 2000, 3050, 2050, 1980],  
    [ 7500, 2050, 2050, 1980],
```

```
[15450, 15050, 15550, 14900]  
]
```

수수료를 0.014 %로 가정할 때, 각 가격에 수수료를 포함한 가격을 한라인에 하나씩 출력하라.

```
2000.28  
3050.427  
2050.2870000000003  
...
```

## 192.

191번의 출력 결과에 행단위로 "----" 구분자를 추가하라.

```
2000.28  
3050.427  
2050.2870000000003  
1980.2772  
----  
7501.05  
2050.2870000000003  
2050.2870000000003  
1980.2772  
----  
15452.163  
15052.107  
15552.177  
14902.086000000001  
----
```

## 193.

192 번 문제의 결괏값을 result 이름의 리스트에 1차원 배열로 저장하라.

```
>> print(result)  
[2000.28,      3050.427,      2050.2870000000003,      1980.2772,      7501.05,
```

2050.2870000000003, 2050.2870000000003, ...]

## 194.

191번 문제의 결괏값을 `result` 이름의 리스트에 2차원 배열로 저장하라. 저장 포맷은 아래와 같다. 각 행에 대한 데이터끼리 리스트에 저장되어야 한다.

```
>> print(result)
[
  [2000.28, 3050.427, 2050.2870000000003, 1980.2772],
  [7501.05, 2050.2870000000003, 2050.2870000000003, 1980.2772],
  [15452.163, 15052.107, 15552.177, 14902.086000000001]
]
```

## 195.

`ohlc` 리스트에는 시가(`open`), 고가 (`high`), 저가 (`low`) , 종가(`close`)가 날짜별로 저장돼 있다. 화면에 종가데이터를 출력하라.

```
ohlc = [{"open", "high", "low", "close"},
        [100, 110, 70, 100],
        [200, 210, 180, 190],
        [300, 310, 300, 310]]
100
190
310
```

## 196.

`ohlc` 리스트에는 시가(`open`), 고가 (`high`), 저가 (`low`) , 종가(`close`)가 날짜별로 저장돼 있다. 종가가 150원보다 큰경우에만 종가를 출력하라.

```
ohlc = [{"open", "high", "low", "close"},
        [100, 110, 70, 100],
        [200, 210, 180, 190],
```

```
[300, 310, 300, 310]]
```

```
190
```

```
310
```

## 197.

- ▶ ohlc 리스트에는 시가(open), 고가 (high), 저가 (low) , 종가(close)가 날짜별로 저장돼 있다. 종가가 시가 보다 크거나 같은 경우에만 종가를 출력하라.

```
ohlc = [{"open", "high", "low", "close"},  
        [100, 110, 70, 100],  
        [200, 210, 180, 190],  
        [300, 310, 300, 310]]
```

```
100
```

```
310
```

## 198.

- ▶ ohlc 리스트에는 시가(open), 고가 (high), 저가 (low) , 종가(close)가 날짜별로 저장돼 있다. 고가와 저가의 차이를 변동폭으로 정의할 때 변동폭을 volatility 이름의 리스트에 저장하라.

```
ohlc = [{"open", "high", "low", "close"},  
        [100, 110, 70, 100],  
        [200, 210, 180, 190],  
        [300, 310, 300, 310]]
```

```
>> print(volatility)
```

```
[40, 30, 10]
```

## 199.

- ▶ 리스트에는 3일 간의 ohlc 데이터가 저장돼 있다. 종가가 시가보다 높은 날의 변동성 (고가 - 저가)을 화면에 출력하라.

```
ohlc = [{"open", "high", "low", "close"},
```

```
[100, 110, 70, 100],  
[200, 210, 180, 190],  
[300, 310, 300, 310]]
```

종가가 시가보다 높은 거래일의 OHLC는 [300, 310, 300, 310] 이다. 따라서 이 거래일의 변동성은 10 (310 - 300)이다.

10

200.

▶ 리스트에는 3일 간의 ohlc 데이터가 저장돼 있다. 시가에 매수해서 종가에 매도했을 경우 총 수익금을 계산하라.

```
ohlc = [{"open", "high", "low", "close"},  
        [100, 110, 70, 100],  
        [200, 210, 180, 190],  
        [300, 310, 300, 310]]
```

1일차 수익 0원 (100 - 100), 2일차 수익 -10원 (190 - 200), 3일차 수익 10원 (310 - 300) 이다.