Week 3. 연습문제 이자 숙제 이자 괴롭힘

1) 9월 초의 네이버 종가는 아래 표와 같습니다. 09/07의 종가를 리스트의 첫 번째 항목으로 입력해서 naver_end_price라는 이름의 리스트를 만들어보세요.

날짜	요일	종가
09/11	금	488,500
09/10	목	500,500
09/09	수	501,000
09/08	화	461,500
09/07	월	474,500

In [2]:
```
naver_end_price = [474500, 461500, 501000, 500500, 488500]
```

2) 문제 1) 에서 만든 naver_end_price를 이용해 해당 주에 종가를 기준으로 가장 높았던 가격을 출력하세요.

(힌트: 리스트에서 최댓값을 찾는 함수는 `max()` 입니다.

In [3]:
```
max_price = max(naver_end_price)
print(max_price)
```

```
501000
```

3) 문제 1) 에서 만든 naver_end_price를 이용해 해당 주에 종가를 기준으로 가장 낮았던 가격을 출력하세요.

(힌트: 리스트에서 최솟값을 찾는 함수는 `min()` 입니다.)

In [4]:
```
min_price = min(naver_end_price)
print(min_price)
```

```
461500
```

4) 문제 1) 에서 만든 naver_end_price를 이용해 해당 주에서 가장 종가가 높았던 요일과 가장 종가가 낮았던 요일의 가격 차를 화면에 출력하세요.

In [5]:
```
diff_price = max_price - min_price
print(diff_price)
```

```
39500
```

5) 문제 1) 에서 만든 naver_end_price를 이용해 수요일의 종가를 화면에 출력하세요.

In [6]:
```
print("수요일 종가: ", naver_end_price[2])
```

```
수요일 종가:  501000
```

6) 문제 1) 의 표를 이용해 날짜를 딕셔너리의 키 값으로, 종가를 딕셔너리의 값으로 사용해 naver_end_price2 라는 딕셔너리를 만드세요.

In [7]:
```
naver_end_price2 = {'09/07':474500,
                    '09/08':461500,
                    '09/09':501000,
                    '09/10':500500,
                    '09/11':488500}
```

7) 문제 6) 에서 만든 naver_end_price2 딕셔너리를 이용해 09/09일의 종가를 출력하세요.

In [8]:
```
print(naver_end_price2['09/09'])
```

```
501000
```

조금 어려운 문제들

1) 다음 리스트 a_list를 5로 나눈 나머지를 기준으로 오름차순으로 정렬 해 보세요.

```
a_list = range(10)
[0, 1, 2, 3, 4, 5, 6, 7, 8, 9]
```

In [15]:
```
a_list = list(range(10))

def mod_5(data):
    remainder = data % 5
    return remainder

a_list.sort(key=mod_5)
print(a_list)
```

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

TA 가 되어 봅시다.

학생들의 정보가 다음과 같다고 생각해봅시다.

```
kevin = {
    "name": "Kevin Kim",
    "homework": [90.0, 97.0, 75.0, 92.0],
    "tests": [75.0, 90.0]
}
alice = {
    "name": "Alice Woo",
    "homework": [100.0, 92.0, 98.0, 100.0],
    "tests": [89.0, 97.0]
}
tyler = {
    "name": "Tyler Swift",
    "homework": [0.0, 87.0, 75.0, 22.0],
    "tests": [100.0, 100.0]
}
```

2) 개개인 학생을 stduent라는 리스트에 모아보세요

In [17]:
```
kevin = {
    "name": "Kevin Kim",
    "homework": [90.0, 97.0, 75.0, 92.0],
    "tests": [75.0, 90.0]
}
alice = {
    "name": "Alice Woo",
    "homework": [100.0, 92.0, 98.0, 100.0],
    "tests": [89.0, 97.0]
}
tyler = {
    "name": "Tyler Swift",
    "homework": [0.0, 87.0, 75.0, 22.0],
    "tests": [100.0, 100.0]
}

students = [kevin, alice, tyler]
```

3) stduent 리스트를 이용하여 모든 학생의 정보를 출력하는 함수 `print_students_info` 를 만들어보세요.
(이 함수는 input인자로 아무것도 받지 않아도 됩니다.)

In [26]:
```
for student in students:
    print('-------student-------')
    for item in student:
        print(item,':',student[item])
```

```
-------student-------
tests : [75.0, 90.0]
name : Kevin Kim
homework : [90.0, 97.0, 75.0, 92.0]
-------student-------
tests : [89.0, 97.0]
name : Alice Woo
homework : [100.0, 92.0, 98.0, 100.0]
-------student-------
tests : [100.0, 100.0]
name : Tyler Swift
homework : [0.0, 87.0, 75.0, 22.0]
```

4) 각 학생을 input 인자로 받아서 `homework` 와 `tests` 의 평균을 return 인자로 돌려주는 함수 `get_average` 를 만들어보세요.

예) `get_average(alice)`

In [33]:
```
def get_average(student):
    homework_avg = sum(student["homework"])/len(student["homework"])
    test_avg = sum(student["tests"])/len(student["tests"])
    return homework_avg, test_avg

get_average(kevin)
```

Out[33]: (88.5, 82.5)

5) 4)에서 만든 함수에서 가중 평균하여 최종 score를 return 하도록 변경 해 보세요.

- homework = 0.3
- test = 0.7

In [52]:
```
def get_average(student):
    weight = {"homework":0.3, "tests":0.7}

    homework_avg = sum(student["homework"])/len(student["homework"])
    test_avg = sum(student["tests"])/len(student["tests"])
    score = homework_avg*weight["homework"] + test_avg*weight["tests"]
    return score

kevin_score = get_average(kevin)
```

6) 최종 score를 이용하여 최종 grade를 return 해 주는 `get_letter_grade` 함수를 만들어보세요.

- 90점 이상: A
- 80점 이상: B
- 70점 이상: C
- 70점 미만: F

In [53]:
```
def get_letter_grade(score):
    if score >= 90:
        return "A"
    elif score >= 80:
        return "B"
    elif score >= 70:
        return "C"
    else:
        return "F"

kevin_score = get_average(kevin)
final_grade_kevin = get_letter_grade(kevin_score)

print('Kevin의 최종 grade: ', final_grade_kevin)
```

```
Kevin의 최종 grade:  B
```

7) 전체 분반의 `homework` 와 `test` 점수의 평균을 print하는 함수 `get_class_avg` 를 만들어보세요.

In [54]:
```
def get_class_avg(students):
    scores = 0
    for student in students:
        scores += get_average(student)

    class_average = scores / len(students)

    return class_average

students = [kevin, alice, tyler]

class_avg = get_class_avg(students)
final_class_grade = get_letter_grade(class_avg)

print(class_avg)
print(final_class_grade)
```

```
87.48333333333333
B
```