

2024 BMED223 :: Hands-on instructions

서동휘 andy1200@korea.ac.kr

박영주 lime2514@korea.ac.kr

문제 지시사항

실습 자료는 블랙보드 '과제 및 시험' 폴더에 업로드 됩니다

제출 지시사항

- 파일명 : 주차번호(week number)_학번_이름.py
예) week07_2024xxxxxx_kildong.py
- 마감기한 : 2024-04-22 23:59
- 지각제출 시 점수 없습니다. 시간 엄수 바랍니다
- 주석(#, ### 등)을 이용하여 문제를 구분하여 작성하시기 바랍니다

Problem 1(*args, **kwargs)

- Pizza topping을 주문하고 이를 수정할 수 있는 함수 2가지를 만들려고 한다.
주문할 수 있는 topping ingredients는 아래와 같다.

Ingredients : "sausage", "pepper", "tomato", "onion", "olives", "mushroom"

(1) **function order_pizza(factor, toppings)**

피자 주문 결과(topping과 그 갯수)를 pizza dictionary로 정리한다.

input:

factor: factor to be multiplied to the number of elements in ingredients.

toppings: part of ingredients. Must be assign more than 0 toppings

e.g. order_pizza(4, "sausage", "pepper")

output:

pizza: dictionary, e.g. {"sausage": 4, "pepper": 4}

(2) **function change_order(pizza, toppings)**

Pizza topping의 종류와 개수를 keyword argument로 받아 pizza dictionary를 변경한다.

input:

pizza: dictionary, e.g. {"sausage": 4, "pepper": 4}

toppings: part of ingredients to be changed. Must be more than 0, e.g. change_ingredients(...

sausage=2, pepper=7)

output:

변경된 pizza dictionary, e.g. {"sausage": 2, "pepper": 7}

code	<code>pizza1 = order_pizza(2, 'sausage', 'pepper', 'tomato', 'tomato', 'onion', 'tomato')</code>
Expected Result	<code>pizza1 = {'sausage': 2, 'pepper': 2, 'tomato': 6, 'onion': 2}</code>
code	<code>change_order(pizza1, sausage=5, onion=3)</code> <code>Print(pizza1)</code>
Expected Result	<code>pizza1 = {'sausage': 5, 'pepper': 2, 'tomato': 6, 'onion': 3}</code>

Problem 2 (list and dictionary as input arguments)

- 병원에서 얻은 patient data를 정리하기 위한 2가지 함수를 아래 지시대로 정의하고 결과를 출력하시오. 이때 원본데이터 patients가 변경되지 않게 코드를 작성해야한다.
- 각각 한 patient에 대한 information은 아래 format의 list로 구성되어 있다.
[height, weight, patient_name, (left vision, right vision), blood_type, age]
- 각 사람의 information을 list로 모아 2차원 list인 patients를 구성하였다.
- ```
patients = [[176, 75, 'Moebius', (0.8, 1.0), 'A', 39],
 [185, 80, 'Riemann', (0.2, 0.3), 'B', 39],
 [176, 72, 'Maxwell', (0.9, 1.0), 'O', 38],
 [178, 72, 'Lagrange', (0.5, 0.6), 'AB', 37],
 [175, 67, 'Laplace', (1.0, 1.1), 'O', 36]]
```

### (1) function change\_name(patients)

- 한 사람의 데이터 중 이름을 3글자만 남기고, 이름 뒤에 "\_{이름의 첫자}"를 붙임.

input: patient list, e.g. patients

output: changed patient list

### (2) function change\_otype(patients)

- 혈액형이 O형인 사람은 마지막 index 뒤에 양쪽 눈의 시력의 산술평균을 추가한다. 이때 산술평균을 구하는 코드는 Lambda function을 정의해서 사용하시오.

input: patient list, e.g. patients

output: changed patient list

| Usage           | change_name(patients)                                                                                                                                                                                                      | change_otype(patients)                                                                                                                                                                                                                            |
|-----------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Expected Result | [[176, 75, 'Moe_M', (0.8, 1.0), 'A', 39],<br>[185, 80, 'Rie_R', (0.2, 0.3), 'B', 39],<br>[176, 72, 'Max_M', (0.9, 1.0), 'O', 38],<br>[178, 72, 'Lag_L', (0.5, 0.6), 'AB', 37],<br>[175, 67, 'Lap_L', (1.0, 1.1), 'O', 36]] | [[176, 75, 'Moebius', (0.8, 1.0), 'A', 39],<br>[185, 80, 'Riemann', (0.2, 0.3), 'B', 39],<br>[176, 72, 'Maxwell', (0.9, 1.0), 'O', 38, 0.95],<br>[178, 72, 'Lagrange', (0.5, 0.6), 'AB', 37],<br>[175, 67, 'Laplace', (1.0, 1.1), 'O', 36, 1.05]] |

## Problem 3

- 2D 4x4 list인 coordinates를 탐색하며 지나간 좌표에 표시를 남기는 robot을 함수로 작성하려고 한다.
- coordinates는 모든 elements를 0로 초기화하여 시작한다.
- move\_robot()에서 지정한 위치(coord)부터 시작
- 로봇은 처음 시작한 좌표에서는 값 변화를 주지 않고, 이동 명령에 따라 도착한 곳에 +1을 해준다.
- list의 범위를 벗어나는 이동 명령은 수행하지 않고, coordinates의 값도 변하지 않는다.
- 0,1,2,3의 이외의 값이 주어지면, 현재 자리의 값을 주어진 값만큼 증가시킨다.
- 필요에 따라 추가로 function을 정의하여 사용하시오.

(1) **function** *move\_robot(coordinates, coord, move)*

input:

- *coordinates*: 원본 및 최종 좌표 list

- *coord*: robot 의 시작 좌표로, tuple(row, column)로 주어진다.

- *moves*: 이동할 방향(0: up, 1: right, 2: down, 3: left)이 갯수 제한 없이(아래 예 참조) 차례대로 주어진다.

output:

- updated 2D list: 이동을 완료한 후, 모든 표시가 완료된 2차원 list

| Test Case | Code Usage                                              | Expected Result                                                |
|-----------|---------------------------------------------------------|----------------------------------------------------------------|
|           | <Initial state> of coordinates                          | [[0, 0, 0, 0]<br>[0, 0, 0, 0]<br>[0, 0, 0, 0]<br>[0, 0, 0, 0]] |
|           | move_robot(coordinates, (2,2), 3,1,0,2,2,2,2)           | [[0, 0, 0, 0]<br>[0, 0, 1, 0]<br>[0, 1, 2, 0]<br>[0, 0, 1, 0]] |
|           | move_robot(coordinates, (0,0), 2,2,2,1,1,1,0,0,0,3,3,3) | [[1, 1, 1, 1]<br>[1, 0, 0, 1]<br>[1, 0, 0, 1]<br>[1, 1, 1, 1]] |

**Problem 4**

- 위에서 만든 *function* 들을 따로 *week07\_package.py* 에 저장하고, 새롭게 만든 *main.py* file 에서 저장한 *package* 를 *call* 하여 아래의 지시사항을 출력하시오.
- *order\_pizza*를 이용하여 새로운 *pizza*를 만들고 두 개 이상의 *topping* 개수를 변경하여 출력하시오.
- *change\_name(patients)*, *change\_otype(patients)*를 수행하고 *patients*를 출력하시오.
- *move\_robot(coordinates, (1,1), 1,5,3,2,1,4,0,0,2)* 의 결과를 출력하시오.

*fin.*