

코딩입문 실습숙제

학과:

학번:

이름:

1. (1) 두 개의 프로그램의 결과를 유추하여 적어보시오. 그 이후 각 코드를 컴퓨터에서 수행하여 그 결과를 확인하시오. (실제 실행 결과는 따로 제출하지 않는다.)

<pre>def func1(): z = 3 print("func1().z: ", z) def func2(x,y): print("func2().z: ", z) return x+y+z z = 10 print("main().z: ", z) func1() print("main().z: ", z) total = func2(4,5) print("func2(): ", total)</pre>	<pre>def func1(): global z z = 3 print("func1().z: ", z) def func2(x,y): print("func2().z: ", z) return x+y+z z = 10 print("main().z: ", z) func1() print("main().z: ", z) total = func2(4,5) print("func2(): ", total)</pre>
---	---

내가 유추한 결과	내가 유추한 결과
<pre>main().z: 10 func1().z: 3 main().z: 10 func2().z: 10 func2(): 19</pre>	<pre>main().z: 10 func1().z: 3 main().z: 3 func2().z: 3 func2(): 12</pre>

(2) 아래 프로그램의 결과를 유추하여 적어보시오. 그 이후 코드를 컴퓨터에서 수행하여 그 결과를 확인하시오. (실제 실행 결과는 따로 제출하지 않는다.)

```
def func3():  
    r = 3  
    g = 4  
    b = 5  
    print(r, g, b)  
  
r, g, b = 0, 1, 2  
  
func3()  
  
print(r, g, b)
```

내가 유추한 결과
3 4 5 0 1 2

- (3) 전역변수 설정과 관련된 코드를 포함하도록 위 프로그램을 수정하면, 두 개의 print 문에서 모두 3, 4, 5 가 출력된다. 해당 코드를 포함한 q1.py 스크립트를 작성하시오.



The image shows a code editor window titled "q1.py — Python_code" with three tabs: q1.py, q4.py, and q5.py. The q1.py tab is active, showing the following Python code:

```
KAU > 13주차 > q1.py > ...  
1  def func3():  
2      global r, g, b  
3      r = 3  
4      g = 4  
5      b = 5  
6  
7      print(r, g, b)  
8  
9  func3()  
10  
11 print(r, g, b)
```

Below the code editor is a terminal window titled "angigyeom — -zsh — 80x24". It shows the command to run the script and its output:

```
[(base) angigyeom@NAVs-MBPR ~ % /usr/local/bin/python3 /Users/angigyeom/Desktop/CODE/Python_code/KAU/13주차/q1.py  
3 4 5  
3 4 5  
(base) angigyeom@NAVs-MBPR ~ %
```

2. 아래 표를 참고하여, 두 문자열 `str_a`, `str_b`를 인자로 받아들이고 문자열 `str_a`과 문자열 `str_b`에서 공통으로 나타나는 문자를 요소로 갖는 리스트를 반환하는 함수 `appear(str_a, str_b)`를 포함하는 `q2.py` script를 작성하시오. 단, 결과 리스트 안의 문자들은 중복되지 않으며, 요소의 순서는 상관없다.

<pre># appear() 정의 부분</pre>	<pre>['a', 'n'] ['b', 'a', 'n'] >>></pre>
<pre># main 프로그램 부분 print(appear('banana', 'na')) print(appear('banana', 'nbza'))</pre>	

```
q2.py — Python_code  
q1.py U q2.py U x q4.py U q5.py U  
KAU > 13주차 > q2.py > ...  
1 def appear(str_a:str, str_b:str) -> any:  
2     string_duplicate = list()  
3     for i in str_a:  
4         for j in str_b:  
5             if i == j and i not in string_duplicate:  
6                 # i에 있는 문자열이랑 j에 있는 문자열이랑 같으면서, 리스트에 해당 문자열이 없  
7                 string_duplicate.append(i)  
8                 # 리스트에 해당 문자열을 append  
9             else:  
10                continue  
11                # 아니라면 continue  
12  
13     return string_duplicate  
14  
15 print(appear('banana', 'na'))  
16 print(appear('banana', 'nbza'))  
  
angigyeom — -zsh — 80x24  
~ — -zsh ... ~ — -zsh  
Last login: Thu Jun  2 23:27:40 on ttys001  
[(base) angigyeom@NAVs-MBPR ~ % /usr/local/bin/python3 /Users/angigyeom/Desktop/C  
ODE/Python_code/KAU/13주차/q2.py  
['a', 'n']  
['b', 'a', 'n']  
(base) angigyeom@NAVs-MBPR ~ %
```

3. 위 2번의 script에서 가장 상위레벨의 반복 횟수를 최소로 하도록 함수 appear(str_a, str_b)를 수정하고, 이를 포함하는 q3.py script를 작성하시오. 예를 들어 appear('banana', 'nbza'), 또는 appear('nbza', 'banana')의 경우, 가장 상위 레벨에서의 반복 단계의 횟수는 6이 아니라 둘다 4가 된다. (6개 문자를 갖는 "banana"를 기준으로

# appear() 정의 부분	['n', 'b', 'a'] # 반복 단계 횟수: 4 ['n', 'b', 'a'] # 반복 단계 횟수: 4 >>>
# main 프로그램 부분 print(appear('banana', 'nbza')) print(appear('nbza', 'banana'))	

다른 문자열을 비교하는 경우, 상위 레벨에서 반복 단계 횟수는 6이다.)

The screenshot shows a Python IDE window titled 'q3.py - Python_code'. The code defines a function 'appear' that takes two strings, 'str_a' and 'str_b', and returns a list of characters. The function uses nested loops to iterate over the characters of 'str_a' and 'str_b'. If a character from 'str_a' is found in 'str_b' and is not already in the 'string_duplicate' list, it is added to the list. The function returns the 'string_duplicate' list. The main program calls 'appear' with 'banana' and 'nbza' as arguments, and prints the result.

```

1 def appear(str_a:str, str_b:str) -> any:
2     string_duplicate = list()
3     if len(str_a) > len(str_b):
4         str_a, str_b = str_b, str_a
5
6     for i in str_a:
7         for j in str_b:
8             if i == j and i not in string_duplicate:
9                 # i에 있는 문자열이랑 j에 있는 문자열이랑 같으면서, 리스트에 해당 문자열이 없
10                string_duplicate.append(i)
11                # 리스트에 해당 문자열을 append
12            else:
13                continue
14            # 아니라면 continue
15
16    return string_duplicate
17
18 print(appear('banana', 'nbza'))
19 print(appear('nbza', 'banana'))

```

The output of the program is shown in a terminal window titled 'angigyeom - zsh - 80x24'. It shows the execution of the script, resulting in the output: ['n', 'b', 'a'] and ['n', 'b', 'a'].

```

(base) angigyeom@NAVs-MBPR ~ % /usr/local/bin/python3 /Users/angigyeom/Desktop/C
ODE/Python_code/KAU/13주 차 /q3.py
['n', 'b', 'a']
['n', 'b', 'a']
(base) angigyeom@NAVs-MBPR ~ %

```

4. 음료를 주문하는 프로그램을 만들고자 한다. 메뉴는 coffee, latte, tea 3가지이고, 가격은 coffee: 2500원, latte: 3000원, tea: 3000원이다. 메뉴를 인자로 받아 해당 금액을 반환하는 price() 함수를 포함하고, 이를 이용하여 아래와 같은 실행결과를 갖는 q4.py script를 작성

```
메뉴) coffee, latte, tea 중에 선택: coffee          # coffee는 사용자 입력데이터
해당 메뉴 개수) 3                                # 3은 사용자 입력데이터
주문내역: ['coffee', 3]
주문총액: 7500
>>>
```

성하시오.

```
q4.py — Python_code
q1.py U q4.py U x q3.py U q5.py U
KAU > 13주차 > q4.py > price > menu_list
1 def price(menu, count):
2     order = list()
3     menu_list = {'coffee': 2500, 'latte': 3000, 'tea': 3000}
4
5     order.append(menu)
6     order.append(count)
7
8     total = menu_list[menu] * count
9
10    print(f"주문내역: {order}")
11    print(f"주문총액: {total}")
12
13
14
15    menu = input("메뉴) coffee, latte, tea 중에 선택: ")
16    count = int(input("해당 메뉴 개수) "))
17
18    price(menu, count)
```

```
angigyeom — -zsh — 80x24
~ — -zsh ... ~ — -zsh ~ — -zsh ~ — -zsh +
Last login: Thu Jun 2 23:29:42 on ttys003
(base) angigyeom@NAVs-MBPR ~ % /usr/local/bin/python3 /Users/angigyeom/Desktop/CODE/Python_code/KAU/13주차/q4.py
메뉴) coffee, latte, tea 중에 선택: coffee
해당 메뉴 개수) 3
주문내역: ['coffee', 3]
주문총액: 7500
(base) angigyeom@NAVs-MBPR ~ %
```

```

주문하실래요? yes/no: y                                # 빨간색은 사용자 입력데이터
yes나 no로만 답해주세요
주문하실래요? yes/no: yes
메뉴) coffee, latte, tea 중 선택: tee
메뉴판에 없는 메뉴입니다. 다시 말씀해주세요
주문하실래요? yes/no: yes
메뉴) coffee, latte, tea 중 선택: tea
해당 메뉴 개수) 1
주문하실래요? yes/no: yes
메뉴) coffee, latte, tea 중 선택: coffee
해당 메뉴 개수) 3
주문하실래요? yes/no: No
yes나 no로만 답해주세요
주문하실래요? yes/no: no
주문내역: [['tea', 1], ['coffee', 3]]
주문총액: 10500
>>>

```

5. 위 4번을 수정하여, 아래 조건을 만족하고 아래 실행결과와 같이 여러 종류의 메뉴를 주문하여 주문내역 리스트와 주문총액을 출력하는 q5.py script를 작성하시오.

- 프로그램은 최소한 아래와 같은 global variable을 갖는다.
 - (1) menu_list: "coffee","latte","tea" 3가지 메뉴를 포함하는 리스트
 - (2) order: 사용자가 입력한 주문리스트로서, 각 요소는 [메뉴, 개수] 형태의 리스트이다. (예시: [[tea, 2], [coffee, 1]])
 - (3) total_price: 사용자의 주문에 따른 총 금액으로 [[tea, 2], [coffee, 1]]의 경우 8500의 정수값을 갖게 된다.
- 프로그램은 먼저 사용자에게 주문 여부("주문하실래요? yes/no: ")를 물어보며, 사용자는 "yes"나 "no" 둘 중에 하나를 정확히 입력해야 한다. 그 외에는 "yes나 no로만 답해주세요"라는 문구를 출력한다. "yes"면 메뉴를 입력하도록 하고, "no"면 현재까지 받은 주문내역과 총액을 출력하고 프로그램을 종료한다.
- 또한, 사용자가 입력한 메뉴를 인자로 받아 메뉴판에 없는 경우 True를, 3가지 메뉴 중의 하나면 False를 반환하는 outof_menu() 함수를 작성하고, 이를 사용하여 메뉴판에 없는 메뉴를 입력하면 "메뉴판에 없는 메뉴입니다. 다시 말씀해주세요"라는 문구를 출력한 뒤 다시 주문할지 여부를 물어본다.
- 한 메뉴와 이에 해당하는 개수를 인자로 받아 전체 주문리스트(order)에 추가하고

```

주문하실래요? yes/no: yes                                # yes는 사용자 입력데이터
메뉴) coffee, latte, tea 중 선택: tea                     # tea는 사용자 입력데이터
해당 메뉴 개수) 2                                         # 2는 사용자 입력데이터
주문하실래요? yes/no: yes                                # yes는 사용자 입력데이터
메뉴) coffee, latte, tea 중 선택: latte                   # latte는 사용자 입력데이터

```

해당 메뉴 개수) 1

주문하실래요? yes/no: no

주문내역: [['tea', 2], ['latte', 1]]

주문총액: 9000

>>>

2는 사용자 입력데이터

no는 사용자 입력데이터

총 금액(total_price)를 업데이트하는 add_order() 함수를 작성하여 사용한다.



```
q5.py — Python_code
q1.py U q4.py U q3.py U q5.py U X
KAU > 13주차 > q5.py > ...
1 # global var: menu_list: list, order: list, total_price: int:
2 # add_order(), outof_order()
3
4
5 menu_list = ["coffee", "latte", "tea"]
6 menu_price = [2500, 3000, 3000]
7 dic_menu_list = {'coffee': 2500, 'latte': 3000, 'tea': 3000}
8 order = list()
9 total_price = 0
10
11
12 def add_order(menu, count):
13     global dic_menu_list, menu_list, menu_price, order, total_price
14     jumunlist = list()
15     jumunlist.append(menu)
16     jumunlist.append(count)
17     order.append(jumunlist)
18     total_price += (count * dic_menu_list[menu])
19
20
21 def outof_menu(menu):
22     global dic_menu_list, menu_list, menu_price, order, total_price
23     while True:
24         if menu == "coffee" or menu == "latte" or menu == "tea":
25             return False
26         else:
27             print('메뉴판에 없는 메뉴입니다. 다시 말씀해주세요.')
28             break
29
30
31 while True:
32     jumun = input('주문하실래요? yes/no: ')
33     if jumun == 'no':
34         print(f'주문내역: {order}\n주문총액:{total_price}')
35         break
36     elif jumun == 'yes':
37         menu = input('메뉴) coffee, latte, tea 중 선택: ')
38         if outof_menu(menu) == False:
39             gaesoo = int(input('해당 메뉴 개수'))
40             add_order(menu, gaesoo)
41         else:
42             continue
43     else:
44         print('yes나 no로만 답해주세요')
```



```
angigyeom — -zsh — 80x24
~ — -zs  ~ — -zsh  ~ — -zsh  ~ — -zsh  ~ — -zsh  +
Last login: Thu Jun  2 23:30:03 on ttys004
(base) angigyeom@NAVs-MBPR ~ % /usr/local/bin/python3 /Users/angigyeom/Desktop/CODE/Python_code/KAU/13주차/q5.py
주문하실래요? yes/no: y
yes나 no로만 답해주세요
주문하실래요? yes/no: yes
메뉴) coffee, latte, tea 중 선택: tee
메뉴판에 없는 메뉴입니다. 다시 말씀해주세요.
주문하실래요? yes/no: yes
메뉴) coffee, latte, tea 중 선택: tea
해당 메뉴 개수)1
주문하실래요? yes/no: yes
메뉴) coffee, latte, tea 중 선택: coffee
해당 메뉴 개수)3
주문하실래요? yes/no: No
yes나 no로만 답해주세요
주문하실래요? yes/no: no
주문내역: [['tea', 1], ['coffee', 3]]
주문총액:10500
(base) angigyeom@NAVs-MBPR ~ %
```