

1. 다음 코드를 람다 함수 형태로 수정할 때, 알맞은 코드를 작성하시오.

```
def f(x, y):  
    return x ** y
```

2. 다음과 같이 리스트 컴프리헨션으로 되어 있는 코드를 람다(lambda) 함수와 map() 함수를 사용하여 표현하시오.

```
>>> ex = [1, 2, 3, 4, 5]  
>>> [value **2 for value in ex]  
[1, 4, 9, 16, 25]
```

3. 다음 코드를 각각 실행하면 서로 다른 결과가 나온다. 이런 결과가 나오는 이유에 대해 서술하시오.

별표(asterisk, \*)는 함수의 인자(argument)로 사용될 때, 시퀀스형 자료형의 각각의 값(element)을 하나의 값(element)으로 넣어주게 된다. 그러므로 'print(\*a)'의 의미는 사실 'print(a[0], a[1], a[2])'으로 나타나게 된다.

```
>>> a = [1, 2, 3]  
>>> print(*a)  
1 2 3  
>>> print(a)  
[1, 2, 3]
```

4. 다음과 같이 코드를 작성했을 때, 예측되는 실행 결과를 쓰시오.

```
>>> def transpose_list(two_dimensional_list):  
...     return [row for row in zip(*two_dimensional_list)]  
...  
>>> transpose_list([[1, 4, 7], [2, 5, 8], [3, 6, 9]])
```

5. 다음 코드의 실행 결과를 쓰시오.

```
>>> date_info = {'year': "2019", 'month': "9", 'day': "6"}
```

```
>>> result = "{year}-{month}-{day}".format(**date_info)
>>> result
```

6. n개의 벡터의 크기가 동일한지 확인하는 함수를 한 줄의 코드로 작성하시오.

7. 다음과 같은 결과를 얻기 위해 하나의 스칼라값을 벡터에 곱하는 코드를 작성하시오.(단 입력되는 벡터의 크기는 일정하지 않음)

```
>>> def scalar_vector_product(alpha, vector_variable):
...
...
>>> scalar_vector_product(5, [1, 2, 3, 4])
[5, 10, 15, 20]
```