Exercise 1

In [56]:

```
def cal(a, b=10, c=None):
    if c is None:
        print(a + b)
    else:
        print(a * b * c)
```

In [58]:

```
cal(5)
```

```
15
```

In [60]:

```
cal(5,3)
```

```
8
```

In [62]:

```
cal(5,3,1)
```

```
15
```

Exercise 2

In [64]:

```
def long(strings):
    return [s for s in strings if len(s) >= 5]
```

In [68]:

```
long(["Apple", "banana", "Cat", "Dog", "Elephant"])
```

Out[68]:

```
['Apple', 'banana', 'Elephant']
```

Exercise 3

In [70]:

```
expression = "3 * 5 + 2"
result = eval(expression)
print(result)
```

```
17
```

Exercise 4

In [82]:

```
def prime(n):
    if n <= 1:
        return False
    for i in range(2, int(n ** 0.5) + 1):
        if n % i == 0:
            return False
    return True

num = [10, 15, 3, 7, 11, 4]
prime_numbers = list(filter(prime, num))
print(prime_numbers)
```

```
[3, 7, 11]
```

Exercise 5

In [84]:

```
strings = ["apple", "banana", "cherry"]
upper = list(map(str.upper, strings))
print(upper)
```

```
['APPLE', 'BANANA', 'CHERRY']
```

Exercise 6

In [86]:

```
strings = ["apple", "banana", "cherry"]
length = list(map(len, strings))
print(length)
```

```
[5, 6, 6]
```

Exercise 7

In [90]:

```
numbers = [1, 2, 3, 4, 5]
total = reduce(lambda x, y: x + y, numbers)
print(total)
```

```
15
```

Exercise 8

In [92]:

```
numbers = [1, 2, 3, 4, 5]
max_value = reduce(lambda x, y: x if x > y else y, numbers)
print(max_value)
```

```
5
```

In [ ]: