

```
def decor(func):  
    def wrapper(a, b):  
        return func(a, b) * 2  
    return wrapper
```

```
def decor2(func):  
    def wrapper(a, b):  
        return func(a, b) - 2  
    return wrapper
```

```
def sum(a, b):  
    return a + b
```

```
fun1 = decor(sum)  
fun2 = decor2(fun1)  
res = fun2(5, 8)  
print(res)
```

```
def decor(func):  
    def wrapper(a, b):  
        return func(a, b) * 2  
    return wrapper
```

```
def decor2(func):  
    def wrapper(a, b):  
        return func(a, b) - 2  
    return wrapper
```

```
@decor2
```

```
@decor
```

```
def sum(a, b):  
    return a + b
```

```
print(sum(4, 5))
```

```
def outer(fun):  
    def wrapper():  
        a = fun()  
        return a + 20  
    return wrapper
```

```
@outer  
def num(): #original  
    return 10  
print(num())
```

```
# num = outer(num)  
# res = num()  
# print(res)
```

```
def outer(*args):  
    def wrapper(func):  
        res = 0  
        for i in args:  
            res += i  
        return res  
    return wrapper
```

```
@outer(4, 5, 6, 7)  
def add(a, b):  
    return a + b
```

```
print(add)
```

```
def outer(func):  
    def wrapper(*args):  
        res = 0  
        for i in args:  
            res += i  
        return res  
    return wrapper
```

```
@outer  
def add(a , b):  
    return a + b
```

```
print(add(3, 4))  
print(add(3, 4, 1, 2))
```

```
def outer(*args, **kwargs):
    def wrapper(func):
        res = 0
        for i in args:
            res += i
        return f"{kwargs['name']} Total marks {res} and percentage {round(res/len(args),2)}"
    return wrapper

@outer(98, 67, 88, 98, 100, 87, name="Rahul")
def add(a, b):
    return a + b

print(add)
```