```
import functools
import logging



def math_eval(func1, func2):
    """Math evaluation decorator.

    Args:
        func1 (_type_): _description_
        func2 (_type_): _description_

    Example:
    >>> @math_eval(ground_truth, generated_func)
    >>> def test_func(x):
    >>>     return x
    >>> result1, result2 = test_func(5)
    >>> print(f"Result from ground_truth: {result1}")
    >>> print(f"Result from generated_func: {result2}")

    """

    def decorator(func):
        @functools.wraps(func)
        def wrapper(*args, **kwargs):
            try:
                result1 = func1(*args, **kwargs)
```

```
            except Exception as e:
                logging.error(f"Error in func1: {e}")
                result1 = None

            try:
                result2 = func2(*args, **kwargs)
            except Exception as e:
                logging.error(f"Error in func2: {e}")
                result2 = None

            if result1 != result2:
                logging.warning(
                    f"Outputs do not match: {result1} != {result2}"
                )

            return result1, result2

        return wrapper

    return decorator


# def ground_truth(x):
#     return x * 2

# def generated_func(x):
```

```
#     return x - 10


# @math_eval(ground_truth, generated_func)

# def test_func(x):

#     return x


# result1, result2 = test_func(5)

# print(f"Result from ground_truth: {result1}")

# print(f"Result from generated_func: {result2}")
```