**Task:**

You are developing a Python utility module that includes a set of mathematical and string-processing functions (e.g., factorial, is\_prime, reverse\_string, normalize\_whitespace). Your goal is to design and implement **comprehensive, maintainable, and readable parameterized test cases** using pytest.

**Requirements:**

1. **Choose 2–4 functions** (math or string utility functions) to test.
2. Write **parameterized test functions** using @pytest.mark.parametrize that:
   * Cover **normal cases**, **edge cases**, and **error cases**
   * Demonstrate **clear organization** and **descriptive test case IDs**
3. Include **at least one test** that uses a **list of tuples** or a **dictionary** for parameter values.
4. Add **comments or documentation** explaining why parameterization is useful for each function.
5. Optionally, use **fixtures** or pytest-cases to show advanced techniques.

**Deliverables:**

* utils.py: The utility module you are testing.
* test\_utils.py: Your pytest test suite with parameterized tests.