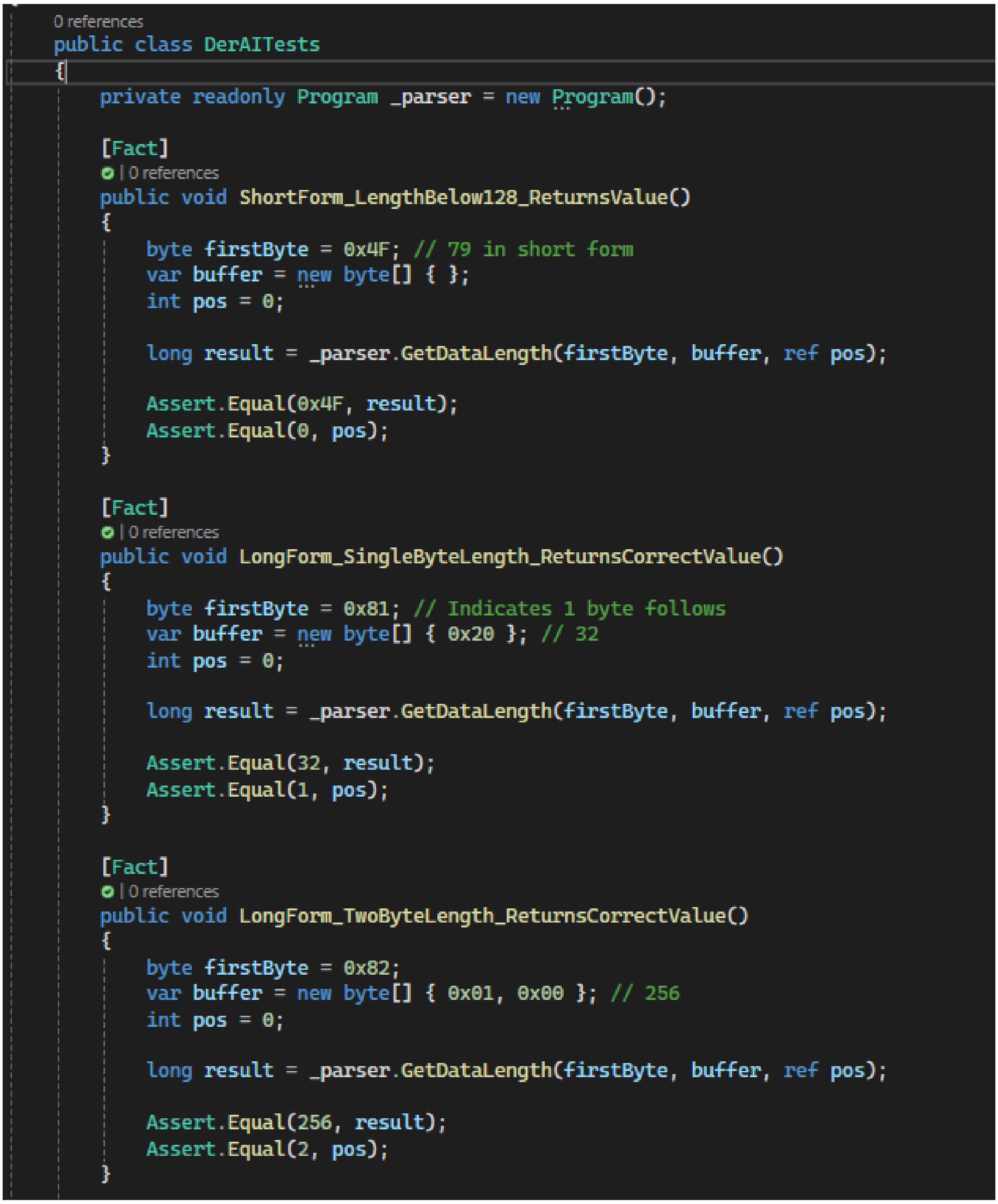
Prompt to ChatGPT :

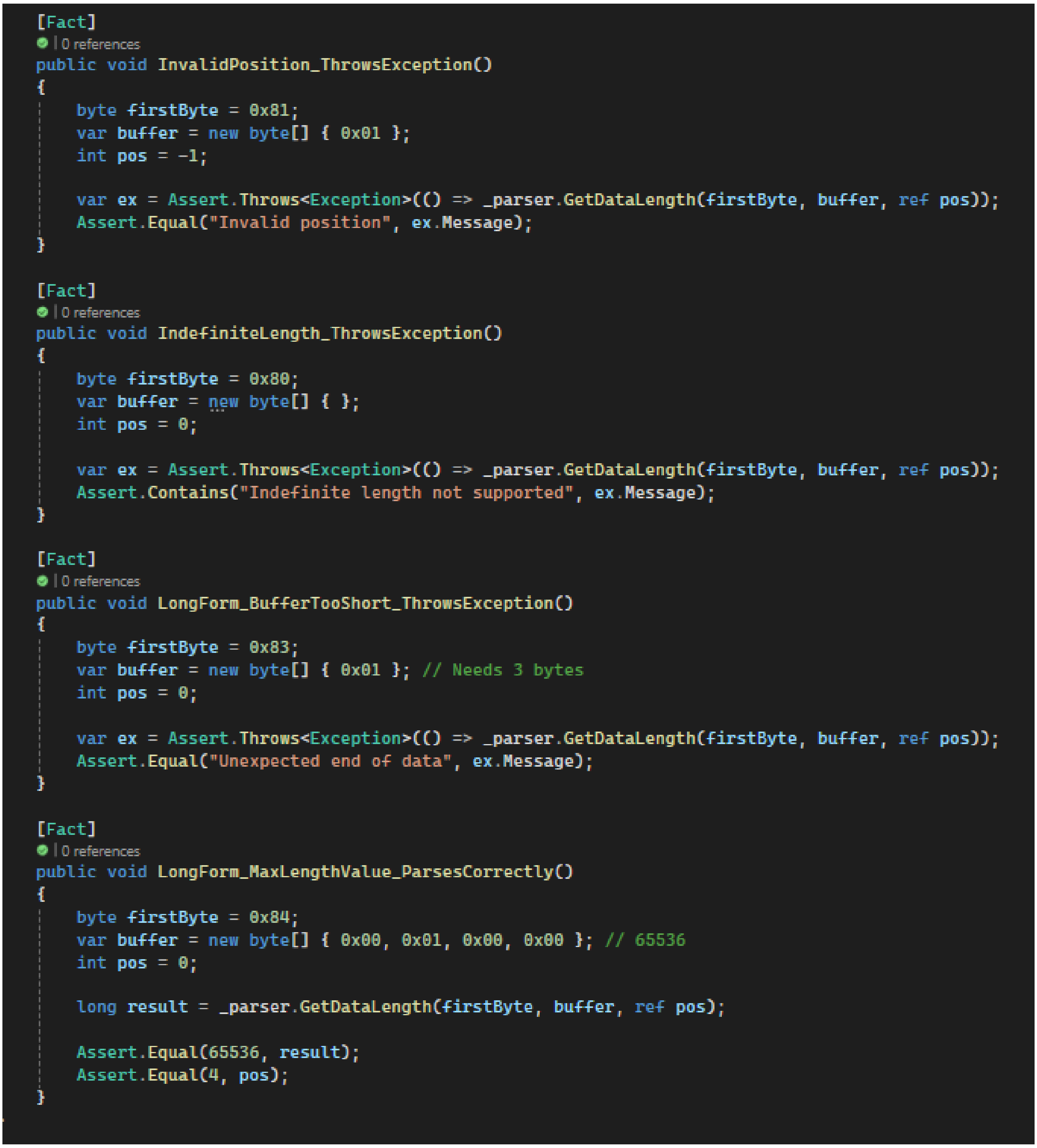
Generate unit tests in C# using xUnit for the method `GetDataLength` from the class below. The method parses DER-encoded lengths (ASN.1) and handles both short and long forms. Include cases that test valid input, edge cases, and error handling.

Function to test : public long GetDataLength(byte firstByte, ReadOnlySpan<byte> buffer, ref int position) { ... (include function body) }

Response

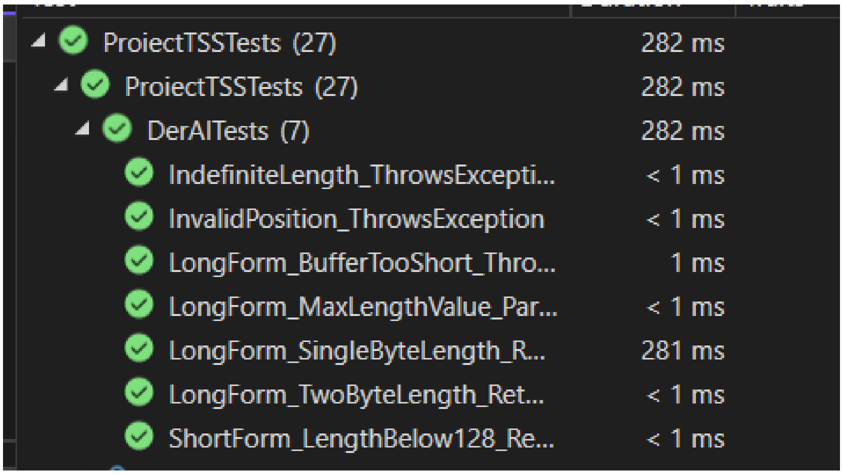


```csharp
public class DerAITests
{
    private readonly Program _parser = new Program();

    [Fact]
    public void ShortForm_LengthBelow128_ReturnsValue()
    {
        byte firstByte = 0x4F; // 79 in short form
        var buffer = new byte[] { };
        int pos = 0;

        long result = _parser.GetDataLength(firstByte, buffer, ref pos);

        Assert.Equal(0x4F, result);
        Assert.Equal(0, pos);
    }

    [Fact]
    public void LongForm_SingleByteLength_ReturnsCorrectValue()
    {
        byte firstByte = 0x81; // Indicates 1 byte follows
        var buffer = new byte[] { 0x20 }; // 32
        int pos = 0;

        long result = _parser.GetDataLength(firstByte, buffer, ref pos);

        Assert.Equal(32, result);
        Assert.Equal(1, pos);
    }

    [Fact]
    public void LongForm_TwoByteLength_ReturnsCorrectValue()
    {
        byte firstByte = 0x82;
        var buffer = new byte[] { 0x01, 0x00 }; // 256
        int pos = 0;

        long result = _parser.GetDataLength(firstByte, buffer, ref pos);

        Assert.Equal(256, result);
        Assert.Equal(2, pos);
    }
```



```
[Fact]
✓ | 0 references
public void InvalidPosition_ThrowsException()
{
    byte firstByte = 0x81;
    var buffer = new byte[] { 0x01 };
    int pos = -1;

    var ex = Assert.Throws<Exception>(() => _parser.GetDataLength(firstByte, buffer, ref pos));
    Assert.Equal("Invalid position", ex.Message);
}

[Fact]
✓ | 0 references
public void IndefiniteLength_ThrowsException()
{
    byte firstByte = 0x80;
    var buffer = new byte[] { };
    int pos = 0;

    var ex = Assert.Throws<Exception>(() => _parser.GetDataLength(firstByte, buffer, ref pos));
    Assert.Contains("Indefinite length not supported", ex.Message);
}

[Fact]
✓ | 0 references
public void LongForm_BufferTooShort_ThrowsException()
{
    byte firstByte = 0x83;
    var buffer = new byte[] { 0x01 }; // Needs 3 bytes
    int pos = 0;

    var ex = Assert.Throws<Exception>(() => _parser.GetDataLength(firstByte, buffer, ref pos));
    Assert.Equal("Unexpected end of data", ex.Message);
}

[Fact]
✓ | 0 references
public void LongForm_MaxLengthValue_ParsesCorrectly()
{
    byte firstByte = 0x84;
    var buffer = new byte[] { 0x00, 0x01, 0x00, 0x00 }; // 65536
    int pos = 0;

    long result = _parser.GetDataLength(firstByte, buffer, ref pos);

    Assert.Equal(65536, result);
    Assert.Equal(4, pos);
}
```

Running autogenerated tests. Success 100% from the first time.



```
◢ ✓ ProiectTSSTests (27)                      282 ms
  ◢ ✓ ProiectTSSTests (27)                    282 ms
    ◢ ✓ DerAITests (7)                        282 ms
        ✓ IndefiniteLength_ThrowsExcepti...   < 1 ms
        ✓ InvalidPosition_ThrowsException     < 1 ms
        ✓ LongForm_BufferTooShort_Thro...       1 ms
        ✓ LongForm_MaxLengthValue_Par...      < 1 ms
        ✓ LongForm_SingleByteLength_R...      281 ms
        ✓ LongForm_TwoByteLength_Ret...       < 1 ms
        ✓ ShortForm_LengthBelow128_Re...      < 1 ms
```

What AI (ChatGPT) Added

New Element	Example	Why It Matters
Test for invalid position	InvalidPosition_ThrowsException → position = -1	Explicitly covers an input validation scenario that was not clearly tested before. Protects against incorrect method calls.
Symbolic values (e.g., 79, 32, 256)	Tests meaningful values, not just boundary cases (e.g., 0, 127)	Improves functional coverage by targeting commonly encountered values in ASN.1 encoding.
Clarity in long-form tests	Separate tests for lengths encoded with 1, 2, 3, and 4 bytes	Makes the test suite more readable and easier to maintain than packing multiple test cases into a single [Theory].
Large but realistic value (65536)	0x84 + 00 01 00 00	Although a test with 0x87 already existed, 65536 is a practical and realistic value for large data buffers.

Conclusion

The tests generated with ChatGPT complemented the existing suite by adding symbolic cases and additional defensive validations, most notably the test for an invalid position. Moreover, the AI provided a clearer structure by generating separate unit tests for each type of case (short form, long form, exceptions), which improved the readability and maintainability of the test code.