```cpp

// COS30008, Midterm, Problem 4, 2024

#include "VigenereForwardIterator.h"

VigenereForwardIterator::VigenereForwardIterator(iVigenereStream& aIStream) :
    fIStream(aIStream),
    fCurrentChar(0),
    fEOF(false)
{
    // Read first character
    fIStream >> fCurrentChar;
}

char VigenereForwardIterator::operator*() const
{
    return fCurrentChar;
}

VigenereForwardIterator& VigenereForwardIterator::operator++()
{
    // Read next character
    fIStream >> fCurrentChar;

    // Check for EOF
    if (!fIStream)
    {
        fEOF = true;
    }

    return *this;
}

VigenereForwardIterator VigenereForwardIterator::operator++(int)
{
    VigenereForwardIterator temp = *this;
    ++(*this);
    return temp;
}

bool VigenereForwardIterator::operator==(const VigenereForwardIterator& aOther) const
{
    // Check if both iterators point to the same stream
    // and have the same EOF status
    return (fIStream == aOther.fIStream)
        && (fEOF == aOther.fEOF);
}
```

```
48
49  bool VigenereForwardIterator::operator!=(const VigenereForwardIterator&
      aOther) const
50  {
51      return !(*this == aOther);
52  }
53
54  VigenereForwardIterator VigenereForwardIterator::begin() const
55  {
56      VigenereForwardIterator temp = *this;
57      // Reset the stream
58      temp.fIStream.reset();
59      return temp;
60  }
61
62  VigenereForwardIterator VigenereForwardIterator::end() const
63  {
64      VigenereForwardIterator temp = *this;
65      // Set EOF status to true
66      temp.fEOF = true;
67      return temp;
68  }
```