```cpp
// Cos3008-Mid
//Created By NUR E SIAM


#include <cctype>
#include "VigenereForwardIterator.h"

VigenereForwardIterator::VigenereForwardIterator(const std::string& aKeyword, const std::string& aSource, EVigenereMode aMode) noexcept : fMode(aMode),
fKeys(aKeyword, aSource),
fSource(aSource),
fIndex(-1),
fCurrentChar('\0')
{
    initializeTable();
}

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

VigenereForwardIterator& VigenereForwardIterator::operator++() noexcept
{
    ++fIndex;
    if (fIndex >= fSource.length())
    {
        return *this;
    }

    if (std::isalpha(fSource[fIndex]))
    {
        if (fMode == EVigenereMode::Encode)
            encodeCurrentChar();
        else
            decodeCurrentChar();

        ++fKeys;
    }
    else
    {
        fCurrentChar = fSource[fIndex]; // Non-alphabetic characters remain unchanged
    }

    return *this;
}

```

```
47  VigenereForwardIterator VigenereForwardIterator::operator++(int) noexcept
48  {
49      VigenereForwardIterator temp = *this;
50      ++(*this);
51      return temp;
52  }
53
54  bool VigenereForwardIterator::operator==(const VigenereForwardIterator&
      aOther) const noexcept
55  {
56      return (fIndex == aOther.fIndex);
57  }
58
59  bool VigenereForwardIterator::operator!=(const VigenereForwardIterator&
      aOther) const noexcept
60  {
61      return !(*this == aOther);
62  }
63
64  VigenereForwardIterator VigenereForwardIterator::begin() const noexcept
65  {
66      return *this;
67  }
68
69  VigenereForwardIterator VigenereForwardIterator::end() const noexcept
70  {
71      VigenereForwardIterator res = *this;
72      res.fIndex = fSource.length();
73      return res;
74  }
75
76  void VigenereForwardIterator::encodeCurrentChar() noexcept
77  {
78      char sourceCharacter = fSource[fIndex];
79      char keyCharacter = std::toupper(*fKeys);
80
81      if (std::isalpha(sourceCharacter))
82      {
83          char encodedCharacter = fMappingTable[std::toupper(keyCharacter) -
              'A'][std::toupper(sourceCharacter) - 'A'];
84          if (std::islower(sourceCharacter))
85          {
86              encodedCharacter = std::tolower(encodedCharacter);
87          }
88          fCurrentChar = encodedCharacter;
89      }
90      else
91      {
92          fCurrentChar = sourceCharacter;
```

```
 93      }
 94  }
 95
 96  void VigenereForwardIterator::decodeCurrentChar() noexcept
 97  {
 98      char sourceCharacter = fSource[fIndex];
 99      char keyCharacter = std::toupper(*fKeys);
100
101      if (std::isalpha(sourceCharacter))
102      {
103          char decodedCharacter = 'A';
104          if (std::isupper(sourceCharacter))
105          {
106              for (int i = 0; i < CHARACTERS; ++i)
107              {
108                  if (fMappingTable[keyCharacter - 'A'][i] == std::toupper
                      (sourceCharacter))
109                  {
110                      decodedCharacter = 'A' + i;
111                      break;
112                  }
113              }
114          }
115          else if (std::islower(sourceCharacter))
116          {
117              for (int i = 0; i < CHARACTERS; ++i)
118              {
119                  if (fMappingTable[keyCharacter - 'A'][i] == std::toupper
                      (sourceCharacter))
120                  {
121                      decodedCharacter = 'a' + i;
122                      break;
123                  }
124              }
125          }
126          fCurrentChar = decodedCharacter;
127      }
128      else
129      {
130          fCurrentChar = sourceCharacter;
131      }
132  }
```