

```
1  #include "Vigenere.h"
2
3  void Vigenere::initializeTable()
4  {
5      for (char row = 0; row < CHARACTERS; row++)
6      {
7          char character = 'B' + row;
8
9          for (char col = 0; col < CHARACTERS; col++)
10         {
11             if (character > 'Z')
12             {
13                 character = 'A';
14             }
15             fMappingTable[row][col] = character++;
16         }
17     }
18 }
19
20 Vigenere::Vigenere(const std::string& aKeyword): fKeyword(aKeyword),      ↗
    fKeywordProvider(KeyProvider(aKeyword))
21 {
22     initializeTable();
23 }
24
25 std::string Vigenere::getCurrentKeyword()
26 {
27     std::string current_keyword;
28
29     for (size_t i = 0; i < fKeyword.length(); i++)
30     {
31         current_keyword += *fKeywordProvider;
32         fKeywordProvider << *fKeywordProvider;
33     }
34     return current_keyword;
35 }
36
37 void Vigenere::reset()
38 {
39     fKeywordProvider.initialize(fKeyword);
40 }
41
42 char Vigenere::encode(char character)
43 {
44     if (isalpha(character))
45     {
46         bool isLower = std::islower(character);
47         char encoded = fMappingTable[*fKeywordProvider - 'A'][std::toupper ↗
            (character) - 'A'];
```

```
48
49     fKeywordProvider << character;
50     if (isLower)
51     {
52         return static_cast<char>(std::tolower(encoded));
53     }
54     return encoded;
55 }
56 return character;
57 }
58
59 char Vigenere::decode(char character)
60 {
61     if (isalpha((character)))
62     {
63         bool isLower = std::islower(character);
64         char encoded = static_cast<char>(toupper(character));
65         char decoded = 0;
66
67         for (char col = 0; col < CHARACTERS; col++)
68         {
69             if (fMappingTable[*fKeywordProvider - 'A'][col] == encoded)
70             {
71                 decoded = static_cast<char>(col + 'A');
72                 break;
73             }
74         }
75
76         fKeywordProvider << decoded;
77         if (isLower)
78         {
79             return static_cast<char>(std::tolower(decoded));
80         }
81         return decoded;
82     }
83     return character;
84 }
```