

Program – 4

AIM: To implement a program to show encryption and decryption through Vigenère's Cipher.

Introduction and Theory

Vigenère's Cipher is a method of encrypting alphabetic text. It uses a simple form of polyalphabetic substitution. A polyalphabetic cipher is any cipher based on substitution, using multiple substitution alphabets. The encryption of the original text is done using the Vigenère's square or Vigenère's table.

- The table consists of the alphabets written out 26 times in different rows, each alphabet shifted cyclically to the left compared to the previous alphabet, corresponding to the 26 possible Caesar Ciphers.
- At different points in the encryption process, the cipher uses a different alphabet from one of the rows.
- The alphabet used at each point depends on a repeating keyword.

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
A	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
B	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A
C	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B
D	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C
E	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D
F	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E
G	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F
H	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G
I	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H
J	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I
K	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J
L	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K
M	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L
N	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M
O	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N
P	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
Q	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P
R	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q
S	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R
T	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S
U	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T
V	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U
W	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V
X	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W
Y	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X
Z	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y

Encryption

The first letter of the plaintext, say G, is paired with A, the first letter of the key. So use row G and column A of the Vigenère square, namely G. Similarly, for the second letter of the plaintext, the second letter of the key is used, the letter at row E and column Y is C. The rest of the plaintext is enciphered in a similar fashion.

Decryption

Decryption is performed by going to the row in the table corresponding to the key, finding the position of the ciphertext letter in this row, and then using the column's label as the plaintext. For example, in row A (from ABC), the ciphertext G appears in column G, which is the first plaintext letter. Next we go to row B (from ABC), locate the ciphertext B which is found in column E, thus E is the second plaintext letter.

Program – 4

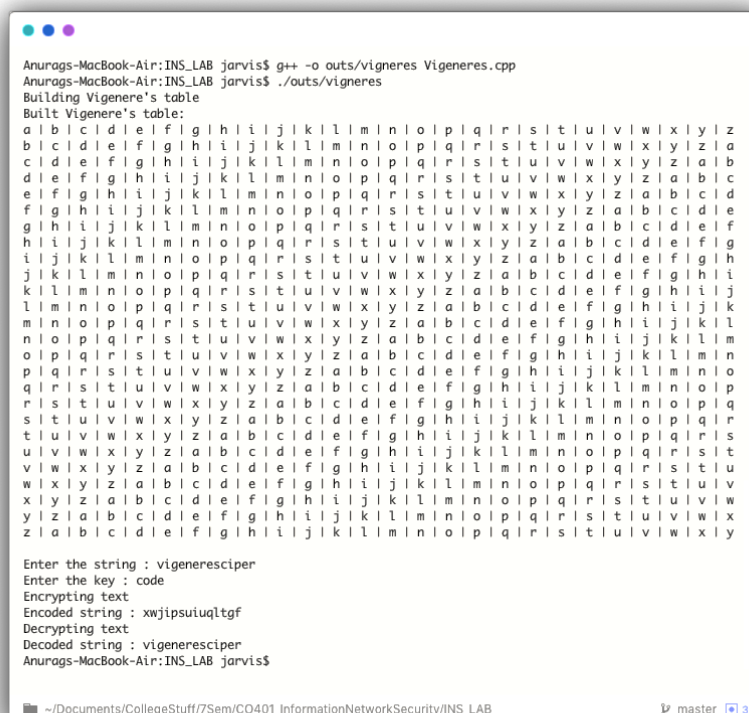
Code

```
1  #include <iostream>
2  #include <string>
3  #include <vector>
4
5  std::vector<std::string> Table;
6
7  void build_table()
8  {
9      for (int i = 0; i < 26; i++)
10     {
11         std::string s = "";
12         for (int j = 0; j < 26; j++)
13         {
14             s += char ((i + j) % 26 + 97);
15         }
16         Table.push_back(s);
17     }
18 }
19
20 int find_index(char s, char k)
21 {
22     for (int i = 0 ; i < 26; i++)
23     {
24         if (Table[int(k-'a')][i] == s)
25             return i;
26     }
27     return -1;
28 }
29
30 void printTable()
31 {
32     for (int i = 0; i < 26; i++)
33     {
34         for (int j = 0; j < 26; j++)
35             std::cout << Table[i][j] << " | ";
36         std::cout << std::endl;
37     }
38 }
39
40 std::string Encrypt(std::string S, std::string key)
41 {
42     std::string enc = "";
43     std::cout << "Encrypting text" << std::endl;
44     for (int i = 0; i < S.length(); i++)
45     {
46         enc += Table[int(key[(i)%key.length()] - 'a')][int(S[i] -
47 'a')];
48     }
49     return enc;
50 }
51
52 std::string Decrypt(std::string S, std::string key)
53 {
54     std::string dec = "";
```

Program – 4

```
55     std::cout << "Decrypting text" << std::endl;
56     for (int i = 0; i < S.length(); i++)
57     {
58         dec += char(find_index(S[i], key[(i%key.length())]) + 97);
59     }
60     return dec;
61 }
62 int main()
63 {
64     std::cout << "Building Vigenere's table" << std::endl;
65     build_table();
66     std::cout << "Built Vigenere's table: " << std::endl;
67     printTable();
68     std::cout << std::endl;
69     std::string input, encoded, decoded, key;
70     std::cout << "Enter the string : ";
71     std::getline(std::cin, input);
72     std::cout << "Enter the key : ";
73     std::cin >> key;
74     encoded = Encrypt(input, key);
75     std::cout << "Encoded string : " << encoded << std::endl;
76     decoded = Decrypt(encoded, key);
77     std::cout << "Decoded string : " << decoded << std::endl;
78     return 0;
79 }
```

Results and Outputs:



```
Anurags-MacBook-Air:INS_LAB jarvis$ g++ -o outs/vigneres Vigeneres.cpp
Anurags-MacBook-Air:INS_LAB jarvis$ ./outs/vigneres
Building Vigenere's table
Built Vigenere's table:
a b c d e f g h i j k l m n o p q r s t u v w x y z
b c d e f g h i j k l m n o p q r s t u v w x y z a
c d e f g h i j k l m n o p q r s t u v w x y z a b
d e f g h i j k l m n o p q r s t u v w x y z a b c
e f g h i j k l m n o p q r s t u v w x y z a b c d
f g h i j k l m n o p q r s t u v w x y z a b c d e
g h i j k l m n o p q r s t u v w x y z a b c d e f
h i j k l m n o p q r s t u v w x y z a b c d e f g
i j k l m n o p q r s t u v w x y z a b c d e f g h
j k l m n o p q r s t u v w x y z a b c d e f g h i
k l m n o p q r s t u v w x y z a b c d e f g h i j
l m n o p q r s t u v w x y z a b c d e f g h i j k
m n o p q r s t u v w x y z a b c d e f g h i j k l
n o p q r s t u v w x y z a b c d e f g h i j k l m
o p q r s t u v w x y z a b c d e f g h i j k l m n
p q r s t u v w x y z a b c d e f g h i j k l m n o
q r s t u v w x y z a b c d e f g h i j k l m n o p
r s t u v w x y z a b c d e f g h i j k l m n o p q
s t u v w x y z a b c d e f g h i j k l m n o p q r
t u v w x y z a b c d e f g h i j k l m n o p q r s
u v w x y z a b c d e f g h i j k l m n o p q r s t
v w x y z a b c d e f g h i j k l m n o p q r s t u
w x y z a b c d e f g h i j k l m n o p q r s t u v
x y z a b c d e f g h i j k l m n o p q r s t u v w
y z a b c d e f g h i j k l m n o p q r s t u v w x
z a b c d e f g h i j k l m n o p q r s t u v w x y

Enter the string : vigenerescipher
Enter the key : code
Encrypting text
Encoded string : xwjipsuiuqltgf
Decrypting text
Decoded string : vigenerescipher
Anurags-MacBook-Air:INS_LAB jarvis$
```

Findings and Learnings:

1. We have implemented Playfair ciphers