**Dharmsinh Desai University, Nadiad**

**Department of Information Technology**

**DAIE, IT704**

**B.Tech. IT, Sem: VII**

**Submitted By**

**Roll No:** IT076

**Name:** Dishant Modh

# Experiment 2: Write a C/C++/Java program to implement Play Fair cipher

# Code

#include <iostream>

#include <string>

using namespace std;

class playfair

{

public:

void doIt( string k, string t, bool ij, bool e )

{

createGrid( k, ij ); getTextReady( t, ij, e );

if( e ) doIt( 1 ); else doIt( -1 );

display();

}

private:

void doIt( int dir )

{

int a, b, c, d; string ntxt;

for( string::const\_iterator ti = \_txt.begin(); ti != \_txt.end(); ti++ )

{

if( getCharPos( \*ti++, a, b ) )

if( getCharPos( \*ti, c, d ) )

{

if( a == c ) { ntxt += getChar( a, b + dir ); ntxt += getChar( c, d + dir ); }

else if( b == d ){ ntxt += getChar( a + dir, b ); ntxt += getChar( c + dir, d ); }

else { ntxt += getChar( c, b ); ntxt += getChar( a, d ); }

}

}

\_txt = ntxt;

}

void display()

{

cout << "\n\n OUTPUT:\n=========" << endl;

string::iterator si = \_txt.begin(); int cnt = 0;

while( si != \_txt.end() )

{

cout << \*si; si++; cout << \*si << " "; si++;

if( ++cnt >= 26 ) cout << endl, cnt = 0;

}

cout << endl << endl;

}

char getChar( int a, int b )

{

return \_m[ (b + 5) % 5 ][ (a + 5) % 5 ];

}

bool getCharPos( char l, int &a, int &b )

{

for( int y = 0; y < 5; y++ )

for( int x = 0; x < 5; x++ )

if( \_m[y][x] == l )

{ a = x; b = y; return true; }

return false;

}

void getTextReady( string t, bool ij, bool e )

{

for( string::iterator si = t.begin(); si != t.end(); si++ )

{

\*si = toupper( \*si ); if( \*si < 65 || \*si > 90 ) continue;

if( \*si == 'J' && ij ) \*si = 'I';

else if( \*si == 'Q' && !ij ) continue;

\_txt += \*si;

}

if( e )

{

string ntxt = ""; size\_t len = \_txt.length();

for( size\_t x = 0; x < len; x += 2 )

{

ntxt += \_txt[x];

if( x + 1 < len )

{

if( \_txt[x] == \_txt[x + 1] ) ntxt += 'X';

ntxt += \_txt[x + 1];

}

}

\_txt = ntxt;

}

if( \_txt.length() & 1 ) \_txt += 'X';

}

void createGrid( string k, bool ij )

{

if( k.length() < 1 ) k = "KEYWORD";

k += "ABCDEFGHIJKLMNOPQRSTUVWXYZ"; string nk = "";

for( string::iterator si = k.begin(); si != k.end(); si++ )

{

\*si = toupper( \*si ); if( \*si < 65 || \*si > 90 ) continue;

if( ( \*si == 'J' && ij ) || ( \*si == 'Q' && !ij ) )continue;

if( nk.find( \*si ) == -1 ) nk += \*si;

}

copy( nk.begin(), nk.end(), &\_m[0][0] );

}

string \_txt; char \_m[5][5];

};

int main( int argc, char\* argv[] )

{

string key, i, txt; bool ij, e;

cout << "(E)ncode or (D)ecode? "; getline( cin, i ); e = ( i[0] == 'e' || i[0] == 'E' );

cout << "Enter a en/decryption key: "; getline( cin, key );

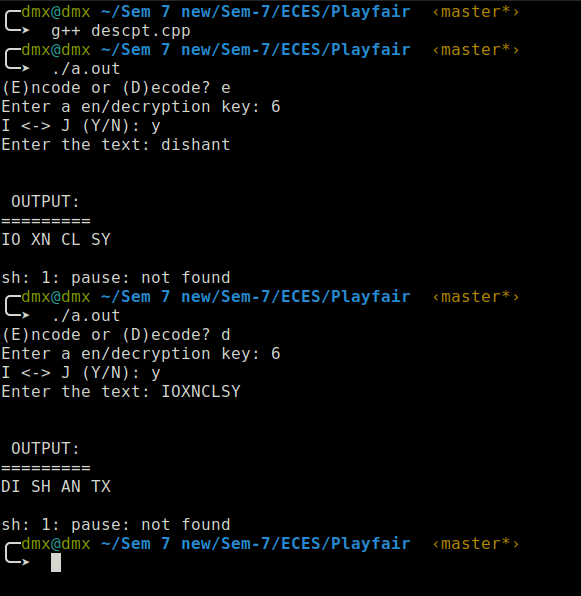
cout << "I <-> J (Y/N): "; getline( cin, i ); ij = ( i[0] == 'y' || i[0] == 'Y' );

cout << "Enter the text: "; getline( cin, txt );

playfair pf; pf.doIt( key, txt, ij, e ); return system( "pause" );

}

1. **Output**

****