```cpp
// Project: General Substitution Ciphers
// Author:  Niko Solihin

#include <iostream>
#include <fstream>
#include <string>
#include <map>

using namespace std;

int main()
{
    // variables needed
    map<char, char> keycode;
    char key[150], value[150];
    string user_input;
    int i = 0, j = 0;

    // allow the user to decide what one wants to do
    int choice;
    cout << "Options\n\n";
    cout << "1. Encryption\n";
    cout << "2. Decryption\n\n";
    cout << "Enter your choice: ";
    cin >> choice;

    // open the key file
    string keyfname = "key.txt";
    ifstream keyfile;
    keyfile.open(keyfname.c_str());
    if (keyfile.fail())
    {
        cout << endl << "Key file (key.txt) not found\n";
        exit(1);
    }
    else
    {
        // key file found, populate array of keys and values
        char ch;
        bool in_bottom_row = false;

        while (keyfile.get(ch))
        {
            if( ch == '\n' )
            {
                in_bottom_row = true;
            }
            if ( !in_bottom_row )
            {
                key[i] = ch;
                i++;
            }
            else
            {
                if ( j < (i+1) )
                {
                    value[j] = ch;
                    j++;
                }
            }
        }
    }

    // open output file
    string outfname = "output.txt";
    ofstream outfile;
    outfile.open (outfname.c_str());

    // open the input file
    string infname;
    cout << "Enter the input file name: ";
    cin >> infname;
    ifstream infile;
    infile.open(infname.c_str());
    if (infile.fail())
    {
     cout << endl << "Input file (p1.txt) not found\n";
     exit(1);
    }
    else
    {
        while( getline(infile, user_input) )
        {
            // get user choice
            if (choice == 1)
            {
                // perform encryption
                for( int m=0; m<i; m++ )
                {
                    keycode[ key[m] ] = value[(m+1)];
                }
            }
            else
            {
                // perform decryption
                for( int n=0; n<i; n++ )
                {
                    keycode[ value[(n+1)] ] = key[n];
                }
                // write to output.txt
            }
            // write to output.txt
            for( int n=0; n<user_input.length(); n++)
            {
                // any character not listed in the key should remain unchanged
                if( keycode.find( user_input[n] ) == keycode.end() )
                {
                    outfile << user_input[n];
                }
                else
                {
                    outfile << keycode[ user_input[n] ];
                }
            }
        }
    }

    infile.close();
    keyfile.close();
    outfile.close();

    cout << "\nOutput.txt generated\n";

    return 0;
}
```