﻿using System;

using Microsoft.VisualStudio.TestTools.UnitTesting;

using SecurityLibrary;

using System.Linq;

namespace SecurityPackageTest

{

[TestClass]

public class PlayfairTest

{

string mainPlain = "armuhsea";

string mainKey = "monoarchy";

string mainCipher = "rmcmbpim".ToUpper();

string mainPlain1 = "hidethegold";

string mainKey1 = "helloworld";

string mainCipher1 = "lfgdnwdpwoav".ToUpper();

string mainPlain2 = "comsecmeanscommunicationssecurity";

string mainPlain22 = "comsecmeanscommunjcatjonssecurjty";

string mainKey2 = "galois";

string mainCipher2 = "dlfdsdndihbddtntuebluoimcvbserulyo".ToUpper();

string mainCipher22 = "dlfdsdndjhbddtntuebluojmcvbserulyo".ToUpper();

string newPlain = "iseeyouthere";

string newKey = "RPMLDSAXICHKQUYEWOZGBFTVN".ToLower();

string newCipher = "CAOSGHZQBQBSOS".ToUpper();

[TestMethod]

public void PlayfairTestEnc1()

{

PlayFair algorithm = new PlayFair();

string cipher = algorithm.Encrypt(mainPlain, mainKey);

Assert.IsTrue(cipher.Equals(mainCipher, StringComparison.InvariantCultureIgnoreCase));

}

[TestMethod]

public void PlayfairTestDec1()

{

PlayFair algorithm = new PlayFair();

string plain = algorithm.Decrypt(mainCipher, mainKey);

Assert.IsTrue(plain.Equals(mainPlain, StringComparison.InvariantCultureIgnoreCase));

}

[TestMethod]

public void PlayfairTestEnc2()

{

PlayFair algorithm = new PlayFair();

string cipher = algorithm.Encrypt(mainPlain1, mainKey1);

Assert.IsTrue(cipher.Equals(mainCipher1, StringComparison.InvariantCultureIgnoreCase));

}

[TestMethod]

public void PlayfairTestDec2()

{

PlayFair algorithm = new PlayFair();

string plain = algorithm.Decrypt(mainCipher1, mainKey1);

Assert.IsTrue(plain.Equals(mainPlain1, StringComparison.InvariantCultureIgnoreCase));

}

[TestMethod]

public void PlayfairTestEnc3()

{

PlayFair algorithm = new PlayFair();

string cipher = algorithm.Encrypt(mainPlain2, mainKey2);

Assert.IsTrue(cipher.Equals(mainCipher2, StringComparison.InvariantCultureIgnoreCase) ||

cipher.Equals(mainCipher22, StringComparison.InvariantCultureIgnoreCase));

}

[TestMethod]

public void PlayfairTestDec3()

{

PlayFair algorithm = new PlayFair();

string plain = algorithm.Decrypt(mainCipher2, mainKey2);

Assert.IsTrue(plain.Equals(mainPlain2, StringComparison.InvariantCultureIgnoreCase) ||

plain.Equals(mainPlain22, StringComparison.InvariantCultureIgnoreCase));

}

string largePlain = "theplayfaircipherusesafivebyfivetablecontainingakeywordorphrasememorizationofthekeywordandfoursimpleruleswasallthatwasrequiredtocreatethefivebyfivetableandusetheciphexlrckhtbrvmbrkhqcrxlrckhtbavheleeatgteenetnwembpqewovtdfheufiknylinthespacesinthetablewiththelettersofthekeyworddroppinganyduplicatelettersthenfilltheremainingspaceswiththerestofthelettersofthealphabetinorderusuallyiandhzittfcsoncapsegteeniohwqdpueityitintfexceruwsoftfdnpelbeoslldhtyvtorightorinsomeotherpatternsuchasaspiralbeginningintheupperlefthandcornerandendinginthecenterthekeywordtogetherwiththeconventionsforfillinginthefivebyfivetableconstitutethecipherkeyxlrckhtbrvmbrkhqcroencryptamessageonewouldbreakthemessageintodigramsgroupsoxlrckhtbemblyvterssuchthatforexamplexlrckhtbrenzloworlxlrckhtbrbecoqrvmbrkhqcrhelloworlxlrckhtbrvmbrkhqcrndmapthemoutonthekeytablxlrckhtbegkmdederxmbrkhqcrppendanuncommonmonogramtocompletethefinaldigraxlrckhtbbmhzetwolettersofthedigramareconsideredastheoppositecornersofarectangleinthekeytablexlrckhtbrctetedrdlwletavosinholohtferooksnrsofthisrectanglxlrckhtbbmhenopdzytiehslzlwrnlgisuurrulexlrckhtbbglwcdplmbrkhqcrtoeachpairoflettersintheplaintextmslxmbrkhqcrfbothlettersarethesamexlrckhtbrcwltvoqenblyvterislefxlrckhtbrvmbrkhqcrddaxlrckhtbrvmbrkhqcrafterthefirstlettexlrckhtbrdkorvsqxtheqewpphbwndboqnftvzmbrkhqcrxlrckhtbrvmbrkhqcrfthelettersappearonthesamerowofyourtablxlrckhtbbvreplacethemwiththeletterstotheirimmediaterightrespectivelyxlrckhtbbvrappingaroundtotheleftsideoftherowifaletterintheoriginalpairwasontherightsideoftheroxlrckhtbbmsmifthelettersappearonthesamecolumnofyourtablexlrckhtbreatorblgeqenmhtfekeyvtersimmediatelybelowrespectivelyxlrckhtbbvrappingaroundtothetopsideofthecolumnifaletterintheoriginalpairwasonthebottomsideofthecolumnmslxmbrkhqcrfthelettersarenotonthesameroworcolumnxlrckhtbreatorblgeqenmhtfekeyvtersonthesamerowrespectivelybutattheotherpairofcornersoftherectangledefinedbytheoriginalpaixlrckhtbbmhzeorderisimportanxlrckhtbbmfeikewmqblyvteroftheencryptedpairistheonethatliesonthesamerowasthefirstletteroftheplaintextpaixlrckhtbrvmbrkhqcrodecryptxlrckhtbeashiegtubearxmbrkhqcrppositexlrckhtbegtfdnowlxmbrkhqcrulesxlrckhtbagshfzmbrkhqcrstasxlrckhtbrvmbrkhqcrdroppinganyextraxlrckhtbrvmbrkhqcrxlrckhtbrvmbrkhqcrxlrckhtbeamhanbokoyuemezsndbittfdhgtanhswsohbahcmkitbslbshsmxlrckhtbbv";

string largeCipher = "NKROMPUIWGDEFWKBFPOBWSGKZDCXGKZDMORNRESTMOHQHQMDTKWCPEEAFRFBSWDTDTPEKYOMKWTSKLKBTKWCPEGDMBKPPFWHLATRFPTRWOSWPMMLGSQOSWDRLYFCRBZEEDDOZKNKRKGYRCUIGYKZSDTRSMRVOBNKREFWKBUNDEFINEDUNDEFINEDUNDEFINEDAKBTRDOMKZKBTKZQSDTRSTCOPZMRGKBPLKFQXQFQLKBWAWDBOHQNKKZSDTRCQNKNKRTKZZKBPPKNKKTCZOPDBBDPAWFMHSMVCPRQFDWZKTRNZZKBPNKBTGKNUMLKBDRVDHQHQHAASERWOKQKNKBDRONPKNKRTKZZKBPPKNKDOURGSCRQKTSDBRDXPVPNUQUGWMBKXKQLKBWSTDWAWDKZKBTKWISMCRPCKQZKQHQLKBZERFPOWPKLKBMORNRKEPNMRKNZXZECFHIZECFXBATKENKRDASNZZKBLPXBISWSWWFDPNRDKHQQHMHHQNKRZSUORFURKNKSMBEPETBDPMBBTCGMHHQNKREBTZKELKBTKWCPEEMAKKZKBCPKQKNKBEWMXBTQKSTPHPEGKNUQFMHHQNKRKGYRCUIGYKZSDTREWXBQKLZZKNKREFWKBEFCZUNDEFINEDUNDEFINEDEKQBCUOLDVBOWSKDSTCOPZMRCDDOTZKBTDBSWSKDHQZECGFDDVAHEPPRWPUNDEFINEDTRNZXZKBPPXBINKOMKPDRVSLATRUNDEFINEDRTXTPOPFUUNDEFINEDCREWTDUNDEFINEDKBNUTPOPFUUNDEFINEDUNDEFINEDMBVDOLKBTAZLSTNKKTCZMORNUNDEFINEDKGTBRBRBUNDEFINEDSUORMBSMXLEWNVTAQNSTAKDPNLWEATRUKZKZKBGKMSMRKHDPUNDEFINEDNKXKZOPTRNZZKBPPKNKRBKHDPVDDREWXBGCRDRBSWNKKESUAPWHZKEWBLRDWPGPDREQSMFMCKQLKBTKZQSDTRUNDEFINEDEZKZKBDRMPQKZDAPWHQKSTPKNKREPETBXBPPKNKHWDREQSMFMUNDEFINEDNKBTSARUZQKBKPNUTPCQMHKPXPFFPTRUNDEFINEDHQPEBRUNDEFINEDZEDOBIASFCPKTRNZZKBPHQNKROMPHQZKZNNANUNDEFINEDHREZFNKZZKBPPDKZKBWSTDUNDEFINEDEPQMZWTBTRNZXZKCFPNRKUNDEFINEDUNDEFINEDBVGDUNDEFINEDUNDEFINEDPGZKELKBGKBPLMKZZKUNDEFINEDBTEDUWNZNKBTCOASFCSMBESTQKLXUNDEFINEDUNDEFINEDUNDEFINEDKLKBTRNZZKBPSAORPDSTNKBODVRDPOPKZWPFMORNUNDEFINEDXDRRUWDKZKBQAKQKNKBTRNZZKBPZENKCKCFNVTDCGOMRDKHKNDRWAREQKZDQUUNDEFINEDXDPSUWFMHPDPZMBZENKRTRKNOGCKEKLKBEPCQGPTRNZZKCFQLKBPEKHHQPMASFCOSWPQLKBCFHINOGCKEKLKBEPUNDEFINEDNANKGNKRTKZZKBPSAORPDSTNKBODVREPTVLTSIUPZELSDTRUNDEFINEDROMPERNKDTCQNKNKRTKZXZKBPGQTDCGOMRTXCRTPODRWAREQKZDQUUNDEFINEDXDPSUWFMHPDPZMBZENKKZPAWHBRPKNKREPTVLQHGPTRNZZKCFQLKBPEKHHQPMASFCOSWPQLKBESNZZENAGCKEKLKBEWUPNQNANUNDEFINEDKLKBTRNZZKBPPDBTEZSTNKBODVRDPOPEEWUPNQUNDEFINEDROMPERNKDTCQNKNKRTKZXZKBPSTNKBODVRDPODRWAREQKZDQURXMONZNKKENKRDASFCPKEWBLRDWPKLKBDREQSMFMRBRKHQRBCXNKKECFHKMSURWGUNDEFINEDNKXKEDBRDHWGQAPELSMUNDEFINEDNKRKFCONTRNZXZKEPKLKBBTEDUWZKRAWGCFONKBSTKZGSLMKCWPQLKBWSTDEPOSONKBGKBPLMKZZKEPKLKBRUWGQLBZLOWGUNDEFINEDUNDEFINEDAERECUOLUNDEFINEDOBNKCKMXRDOBUNDEFINEDSUAPWHZKUNDEFINEDKLKBMPONUNDEFINEDPUBOUNDEFINEDMBNKUNDEFINEDONSWUNDEFINEDUNDEFINEDBDPAWFMHSMZCZNDPUNDEFINEDUNDEFINEDUNDEFINEDUNDEFINEDUNDEFINEDONGSMESTEZVDTKOBXBCKQLKBGKMSMNBOWSKDSIBTGKQHBNRBNANUNDEFINEDX";

string largePlainForAnlysis = "THEPLAYFAIRCIPHERUSESAFIVEBYFIVETABLECONTAININGAKEYWORDORPHRASEMEMORIZATIONOFTHEKEYWORDANDFOURSIMPLERULESWASALLTHATWASREQUIREDTOCREATETHEFIVEBYFIVETABLEANDUSETHECIPHEXLRCKHTBRVMBRKHQCRXLRCKHTBAVHELEEATGTEENETNWEMBPQEWOVTDFHEUFIKNYLINTHESPACESINTHETABLEWITHTHELETTERSOFTHEKEYWORDDROPPINGANYDUPLICATELETXTERSTHENFILXLTHEREMAININGSPACESWITHTHERESTOFTHELETTERSOFTHEALPHABETINORDERUSUALXLYIANDHZITTFCSONCAPSEGTEENIOHWQDPUEITYITINTFEXCERUWSOFTFDNPELBEOSLLDHTYVTORIGHTORINSOMEOTHERPATXTERNSUCHASASPIRALBEGINNINGINTHEUPXPERLEFTHANDCORNERANDENDINGINTHECENTERTHEKEYWORDTOGETHERWITHTHECONVENTIONSFORFILXLINGINTHEFIVEBYFIVETABLECONSTITUTETHECIPHERKEYXLRCKHTBRVMBRKHQCROENCRYPTAMESSAGEONEWOULDBREAKTHEMESXSAGEINTODIGRAMSGROUPSOXLRCKHTBEMBLYVTERSSUCHTHATFOREXAMPLEXLRCKHTBRENZLOWORLXLRCKHTBRBECOQRVMBRKHQCRHELXLOWORLXLRCKHTBRVMBRKHQCRNDMAPTHEMOUTONTHEKEYTABLXLRCKHTBEGKMDEDERXMBRKHQCRPXPENDANUNCOMXMONMONOGRAMTOCOMPLETETHEFINALDIGRAXLRCKHTBBMHZETWOLETXTERSOFTHEDIGRAMARECONSIDEREDASTHEOPXPOSITECORNERSOFARECTANGLEINTHEKEYTABLEXLRCKHTBRCTETEDRDLWLETAVOSINHOLOHTFEROOKSNRSOFTHISRECTANGLXLRCKHTBBMHENOPDZYTIEHSLZLWRNLGISUURRULEXLRCKHTBBGLWCDPLMBRKHQCRTOEACHPAIROFLETXTERSINTHEPLAINTEXTMSLXMBRKHQCRFBOTHLETTERSARETHESAMEXLRCKHTBRCWLTVOQENBLYVTERISLEFXLRCKHTBRVMBRKHQCRDXDAXLRCKHTBRVMBRKHQCRAFTERTHEFIRSTLETTEXLRCKHTBRDKORVSQXTHEQEWPPHBWNDBOQNFTVZMBRKHQCRXLRCKHTBRVMBRKHQCRFTHELETXTERSAPPEARONTHESAMEROWOFYOURTABLXLRCKHTBBVREPLACETHEMWITHTHELETXTERSTOTHEIRIMXMEDIATERIGHTRESPECTIVELYXLRCKHTBBVRAPXPINGAROUNDTOTHELEFTSIDEOFTHEROWIFALETXTERINTHEORIGINALPAIRWASONTHERIGHTSIDEOFTHEROXLRCKHTBBMSMIFTHELETTERSAPPEARONTHESAMECOLUMNOFYOURTABLEXLRCKHTBREATORBLGEQENMHTFEKEYVTERSIMMEDIATELYBELOWRESPECTIVELYXLRCKHTBBVRAPXPINGAROUNDTOTHETOPSIDEOFTHECOLUMNIFALETXTERINTHEORIGINALPAIRWASONTHEBOTXTOMSIDEOFTHECOLUMNMSLXMBRKHQCRFTHELETXTERSARENOTONTHESAMEROWORCOLUMNXLRCKHTBREATORBLGEQENMHTFEKEYVTERSONTHESAMEROWRESPECTIVELYBUTATXTHEOTHERPAIROFCORNERSOFTHERECTANGLEDEFINEDBYTHEORIGINALPAIXLRCKHTBBMHZEORDERISIMPORTANXLRCKHTBBMFEIKEWMQBLYVTEROFTHEENCRYPTEDPAIRISTHEONETHATLIESONTHESAMEROWASTHEFIRSTLETTEROFTHEPLAINTEXTPAIXLRCKHTBRVMBRKHQCRODECRYPTXLRCKHTBEASHIEGTUBEARXMBRKHQCRPXPOSITEXLRCKHTBEGTFDNOWLXMBRKHQCRULESXLRCKHTBAGSHFZMBRKHQCRSTASXLRCKHTBRVMBRKHQCRDROPPINGANYEXTRAXLRCKHTBRVMBRKHQCRXLRCKHTBRVMBRKHQCRXLRCKHTBEAMHANBOKOYUEMEZSNDBITTFDHGTANHSWSOHBAHCMKITBSLBSHSMXLRCKHTBBV".ToLower();

string largeKey = "pasword";

[TestMethod]

public void PlayfairTestEnc4()

{

PlayFair algorithm = new PlayFair();

string cipher = algorithm.Encrypt(largePlain, largeKey);

Assert.IsTrue(cipher.Equals(largeCipher, StringComparison.InvariantCultureIgnoreCase));

}

[TestMethod]

public void PlayfairTestDec4()

{

PlayFair algorithm = new PlayFair();

string plain = algorithm.Decrypt(largeCipher, largeKey);

Assert.IsTrue(plain.Equals(largePlain, StringComparison.InvariantCultureIgnoreCase));

}

[TestMethod]

public void PlayfairTestBonusAnalysis()

{

PlayFair algorithm = new PlayFair();

string plain = algorithm.Analyse(largeCipher);

int count = Enumerable.Range(0, largePlainForAnlysis.Length)

.Count(i => largePlainForAnlysis[i] == plain[i]);

Assert.IsTrue(count \* 100 / largePlain.Length > 50);

}

[TestMethod]

public void PlayfairTestNewEnc()

{

PlayFair algorithm = new PlayFair();

string cipher = algorithm.Encrypt(newPlain, newKey);

Assert.IsTrue(cipher.Equals(newCipher, StringComparison.InvariantCultureIgnoreCase));

}

[TestMethod]

public void PlayfairTestNewDec()

{

PlayFair algorithm = new PlayFair();

string plain = algorithm.Decrypt(newCipher, newKey);

Assert.IsTrue(plain.Equals(newPlain, StringComparison.InvariantCultureIgnoreCase));

}

}

}