Walchand College Of Engineering, Sangli

Department of Computer Science and Engineering

Subject: C&NS Lab

Batch: B4

Name: Gayatri Sopan Gade PRN:2020BTECS00210

Assignment 3

Title: Implementation of Play Fair Cipher

Introduction:

The Playfair cipher was the first practical digraph substitution cipher. The scheme was invented in 1854 by Charles Wheatstone but was named after Lord Playfair who promoted the use of the cipher. In playfair cipher unlike traditional cipher we encrypt a pair of alphabets(digraphs) instead of a single alphabet.
It was used for tactical purposes by British forces in the Second Boer War and in World War I and for the same purpose by the Australians during World War II. This was because Playfair is reasonably fast to use and requires no special equipment.

Algorithm:

1. **Generate the key Square(5×5):**

- The key square is a 5×5 grid of alphabets that acts as the key for encrypting the plaintext. Each of the 25 alphabets must be unique and one letter of the alphabet (usually J) is omitted from the table (as the table can hold only 25 alphabets). If the plaintext contains J, then it is replaced by I.

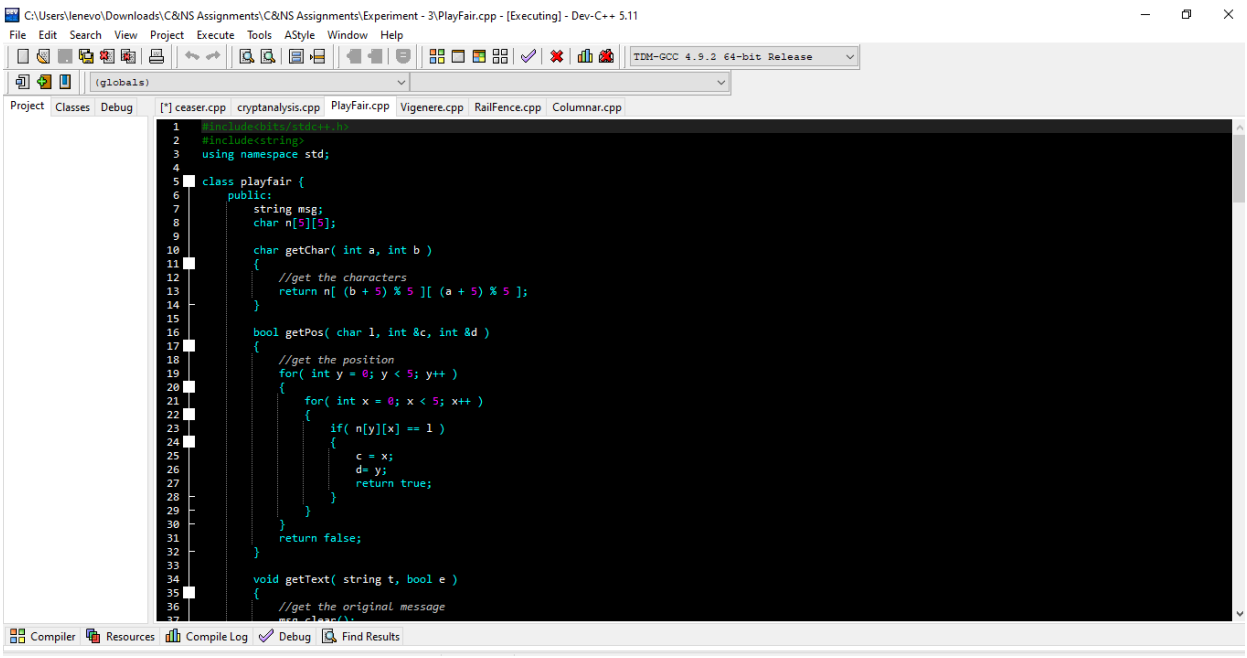
- The initial alphabets in the key square are the unique alphabets of the key in the order in which they appear followed by the remaining letters of the alphabet in order.

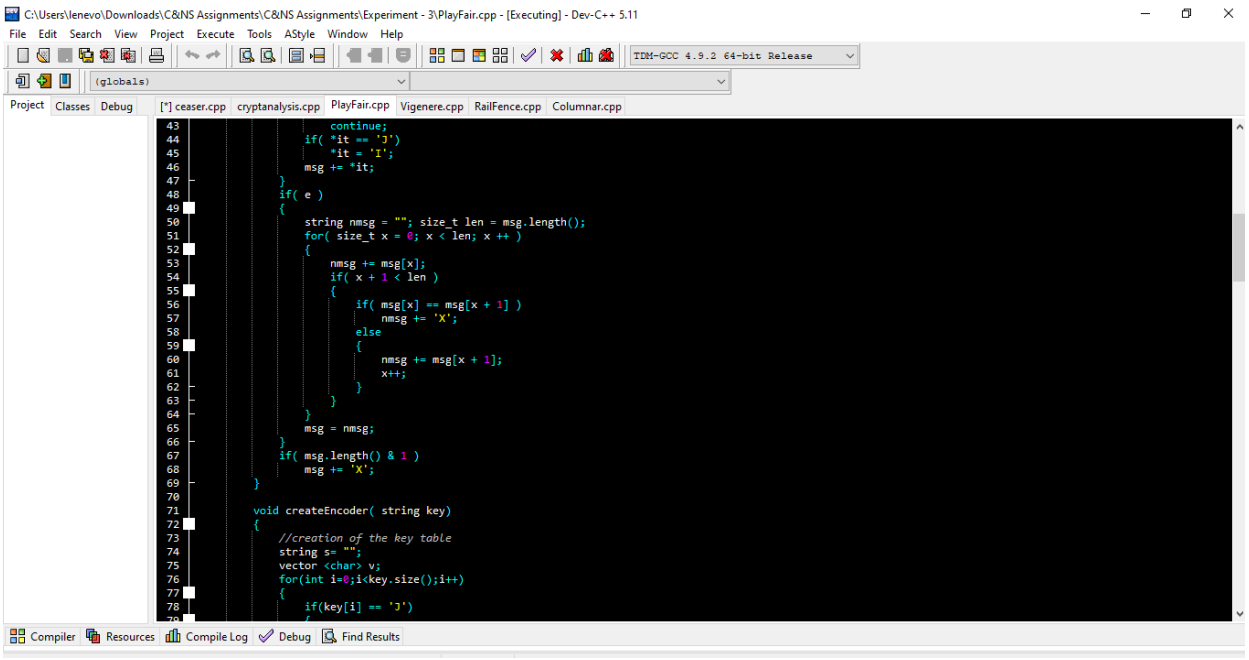
2. **Algorithm to encrypt the plain text:** The plaintext is split into pairs of two letters. If there is an odd number of letters, a Z is added to the last letter.

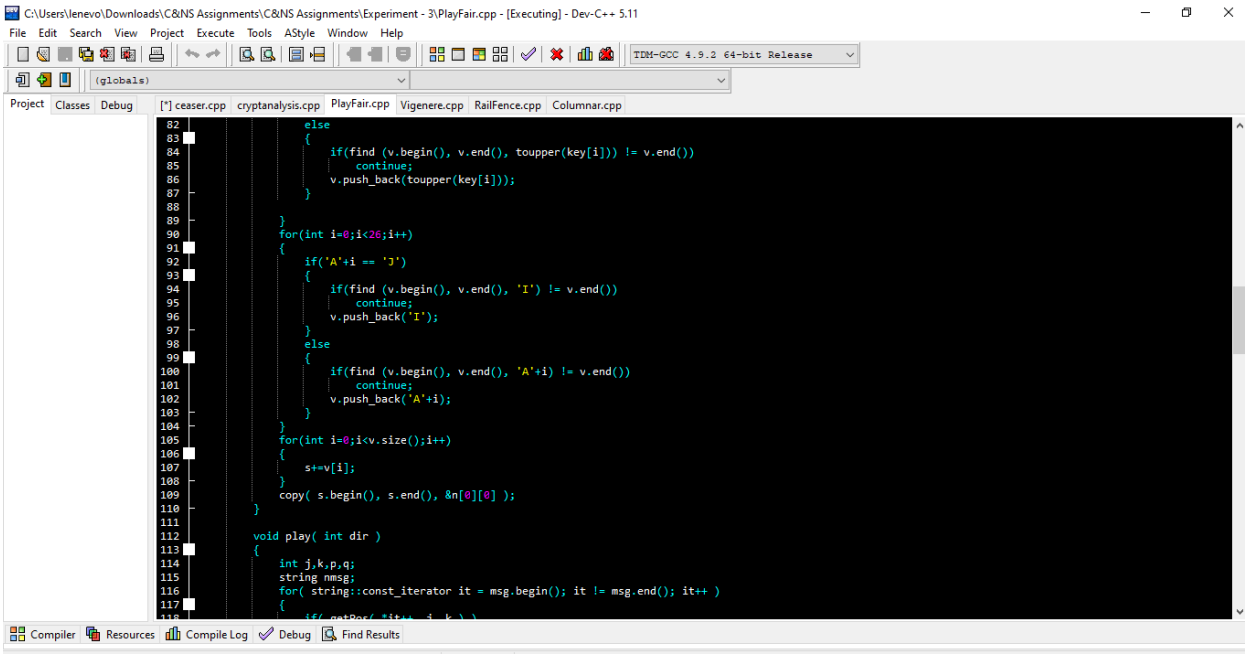
Rules:

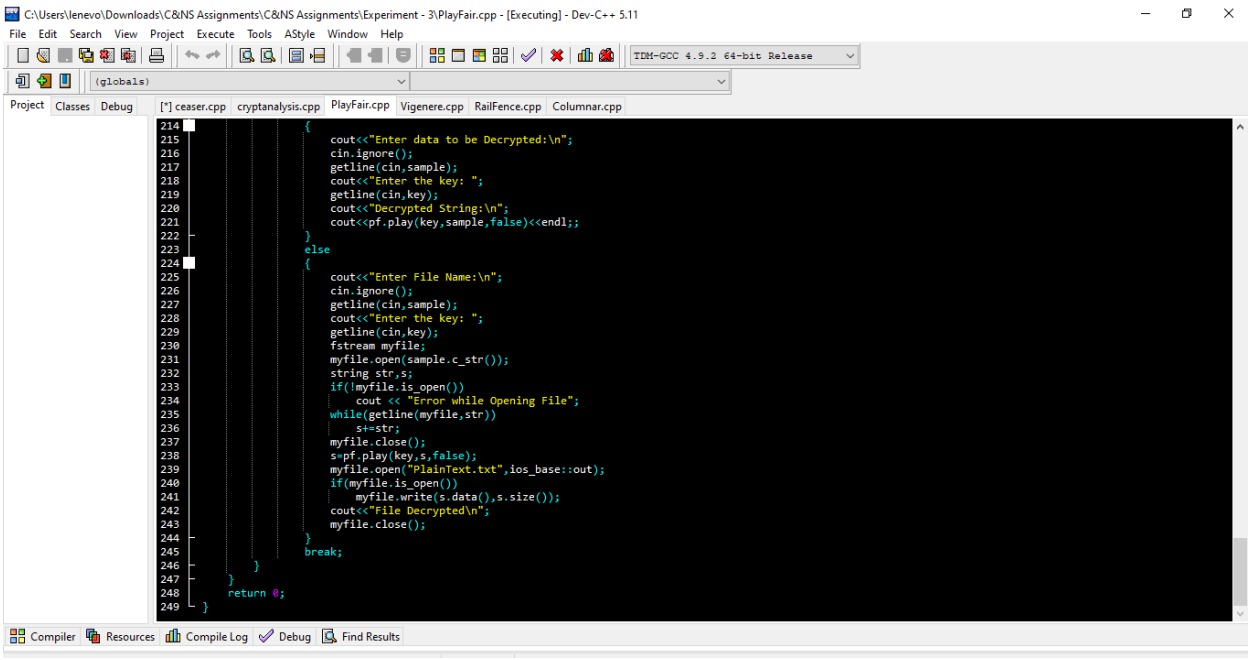
- If both the letters are in the same column: Take the letter below each one (going back to the top if at the bottom)
- If both the letters are in the same row: Take the letter to the right of each one (going back to the leftmost if at the rightmost position)
- If neither of the above rules is true: Form a rectangle with the two letters and take the letters on the horizontal opposite corner of the rectangle.

Code:

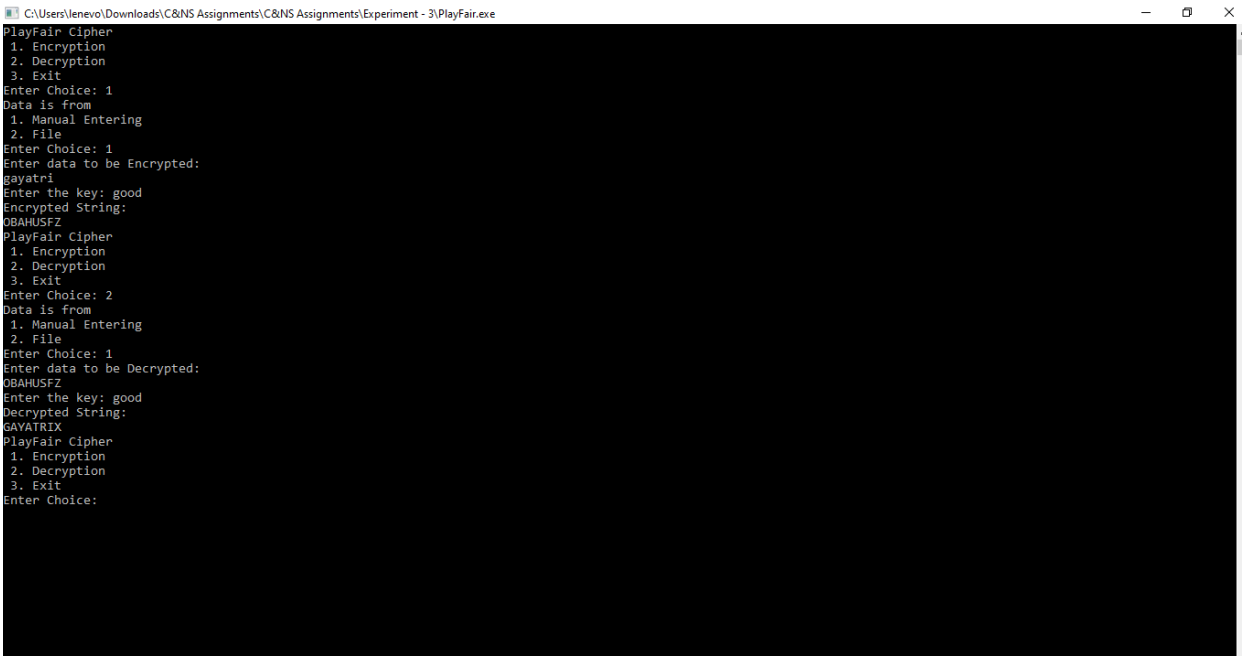



C:\Users\lenevo\Downloads\C&NS Assignments\C&NS Assignments\Experiment - 3\PlayFair.cpp - [Executing] - Dev-C++ 5.11
File Edit Search View Project Execute Tools AStyle Window Help
TDM-GCC 4.9.2 64-bit Release
(globals)
Project Classes Debug
[*] ceaser.cpp cryptanalysis.cpp PlayFair.cpp Vigenere.cpp RailFence.cpp Columnar.cpp
continue;
if(*it == 'J')
*it = 'I';
msg += *it;
}
if(e)
{
string nmsg = ""; size_t len = msg.length();
for(size_t x = 0; x < len; x ++)
{
nmsg += msg[x];
if(x + 1 < len)
{
if(msg[x] == msg[x + 1])
nmsg += 'X';
else
{
nmsg += msg[x + 1];
x++;
}
}
}
msg = nmsg;
}
if(msg.length() & 1)
msg += 'X';
}
void createEncoder(string key)
{
//creation of the key table
string s= "";
vector <char> v;
for(int i=0;i<key.size();i++)
{
if(key[i] == 'J')
Compiler Resources Compile Log Debug Find Results


C:\Users\lenevo\Downloads\C&NS Assignments\C&NS Assignments\Experiment - 3\PlayFair.cpp - [Executing] - Dev-C++ 5.11
File Edit Search View Project Execute Tools AStyle Window Help
TDM-GCC 4.9.2 64-bit Release
(globals)
Project Classes Debug
[*] ceaser.cpp cryptanalysis.cpp PlayFair.cpp Vigenere.cpp RailFence.cpp Columnar.cpp
else
{
if(find (v.begin(), v.end(), toupper(key[i])) != v.end())
continue;
v.push_back(toupper(key[i]));
}
}
for(int i=0;i<26;i++)
{
if('A'+i == 'J')
{
if(find (v.begin(), v.end(), 'I') != v.end())
continue;
v.push_back('I');
}
else
{
if(find (v.begin(), v.end(), 'A'+i) != v.end())
continue;
v.push_back('A'+i);
}
}
for(int i=0;i<v.size();i++)
{
s+=v[i];
}
copy(s.begin(), s.end(), &n[0][0]);
}
void play(int dir)
{
int j,k,p,q;
string nmsg;
for(string::const_iterator it = msg.begin(); it != msg.end(); it++)
{
Compiler Resources Compile Log Debug Find Results


C:\Users\lenevo\Downloads\C&NS Assignments\C&NS Assignments\Experiment - 3\PlayFair.cpp - [Executing] - Dev-C++ 5.11
File Edit Search View Project Execute Tools AStyle Window Help
TDM-GCC 4.9.2 64-bit Release
(globals)
Project Classes Debug
[*] ceaser.cpp cryptanalysis.cpp PlayFair.cpp Vigenere.cpp RailFence.cpp Columnar.cpp
{
cout<<"Enter data to be Decrypted:\n";
cin.ignore();
getline(cin,sample);
cout<<"Enter the key: ";
getline(cin,key);
cout<<"Decrypted String:\n";
cout<<pf.play(key,sample,false)<<endl;;
}
else
{
cout<<"Enter File Name:\n";
cin.ignore();
getline(cin,sample);
cout<<"Enter the key: ";
getline(cin,key);
fstream myfile;
myfile.open(sample.c_str());
string str,s;
if(!myfile.is_open())
cout << "Error while Opening File";
while(getline(myfile,str))
s+=str;
myfile.close();
s=pf.play(key,s,false);
myfile.open("PlainText.txt",ios_base::out);
if(myfile.is_open())
myfile.write(s.data(),s.size());
cout<<"File Decrypted\n";
myfile.close();
}
break;
}
}
return 0;
}
Compiler Resources Compile Log Debug Find Results

Output:


C:\Users\lenevo\Downloads\C&NS Assignments\C&NS Assignments\Experiment - 3\PlayFair.exe
PlayFair Cipher
1. Encryption
2. Decryption
3. Exit
Enter Choice: 1
Data is from
1. Manual Entering
2. File
Enter Choice: 1
Enter data to be Encrypted:
gayatri
Enter the key: good
Encrypted String:
OBAHUSFZ
PlayFair Cipher
1. Encryption
2. Decryption
3. Exit
Enter Choice: 2
Data is from
1. Manual Entering
2. File
Enter Choice: 1
Enter data to be Decrypted:
OBAHUSFZ
Enter the key: good
Decrypted String:
GAYATRIX
PlayFair Cipher
1. Encryption
2. Decryption
3. Exit
Enter Choice: