DISCREATE MATHEMATICS

PROJECT

## Project Title:

**Playfair Cipher Project Members:**

|  |  |
| --- | --- |
| **Student Name** | **Student ID** |
| Muzamil Khan | **11508** |
| Zain Zahid | **11515** |
| Ansar Ali |  |

## Playfair 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](https://www.geeksforgeeks.org/caesar-cipher/) we encrypt a pair of alphabets(digraphs) instead of a single alphabet.

## Playfair Cipher Use:

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.

**Techopedia Explains Playfair Cipher:**

Created by Sir Charles Wheatstone, the Playfair cipher is named for Lord Playfair, who promoted its use. It was later used by the British and Australians during World War I and World War II.

The Playfair cipher's essential method involves creating key tables that arrange the letters of the alphabet into a square grid. With these key tables, the user separates the text of a message into two-letter bits. To encode the message, each two-letter bit is transposed on the key table. Thus, to decode the message, the receiver requires the key table itself.

As an example of the Playfair cipher, begin with the following message text: HELLO WORLD. After splitting this message into two-letter bits, a user starts with the letters HE and LL. If the letter H is in the top left corner and the letter E is in the bottom right corner of the key table, the encoded two-letter bit, which consists of the letters in the top right corner and bottom left corner, is transposed on the key table. This "mirror" technique enables effective text encoding without any equipment or infrastructure, except for printed text on paper.

These types of ciphers constituted the most advanced encryption for the time period before modern computing emerged. By contrast, the most advanced encryption techniques of the digital era involve computer algorithms, rather than this type of labor-intensive manual encryption.

# Advantages and Disadvantages:

### Advantages:

* 1. It is significantly harder to break since the frequency analysis technique used to break simple substitution ciphers is difficult but still can be used on (25\*25) = 625 digraphs rather than 25 monographs which is difficult.
  2. Frequency analysis thus requires more cipher text to crack the encryption.

### Disadvantages:

1. An interesting weakness is the fact that a digraph in the ciphertext (AB) and it’s reverse (BA) will have corresponding plaintexts like UR and RU (and also ciphertext UR and RU will correspond to plaintext AB and BA, i.e.

the substitution is self-inverse). That can easily be exploited with the aid of frequency analysis, if the language of the plaintext is known.

1. Another disadvantage is that playfair cipher is a [symmetric cipher](https://www.geeksforgeeks.org/traditional-symmetric-ciphers/) thus same key is used for both encryption and decryption.

# PYTHON CODING:

import math

def encrypt(mesag, ekey): word = mesag

list\_of\_letters = list(word) numbers = []

for letter in list\_of\_letters: number = ord(letter) - 97 if number == -65:

numbers.append(number) else:

f=math.ceil(math.fmod(number+(ekey),26)) numbers.append(f)

lett = []

for numb in numbers: lette= chr(numb+97) lett.append(lette)

strmessage = "".join(lett)

print("YOUR Encrypt MESSAGE IS :" , strmessage)

def decrypt(meesg, dkey): dword = meesg

d\_list\_of\_letters = list(dword) dnumbers = []

for dletter in d\_list\_of\_letters: dnumber = ord(dletter) - 97 if dnumber == -65:

dnumbers.append(dnumber) elif dnumber > -(dkey):

df=math.ceil(math.fmod((dnumber+26)-(dkey),26)) dnumbers.append(df)

else:

df=math.ceil(math.fmod(dnumber-(dkey),26)) dnumbers.append(df)

dlett = []

for dnumb in dnumbers: dlette= chr(dnumb+97) dlett.append(dlette)

dstrmessage = "".join(dlett)

print("YOUR Decrypt MESSAGE IS :", dstrmessage)

while True:

choies=int (input("\n\nPLESE SELECT YOUR CHOISE\n\nPRESS 1 FOR Encryption\nPRESS 2 FOR Decryption\nPRESS 3 FOR EXIT\nyour choise :"))

if choies == 1:

enke=int (input("ENTER KEY FOR Encryption:")) enmessage=input("PLEASE ENTER YOUR MESSAGE IN SMALL LETTERS:

")

encrypt(enmessage, enke) elif choies ==2:

dyke=int (input("ENTER KEY FOR Dycryption:")) dymessage=input("PLEASE ENTER YOUR MESSAGE IN SMALL LETTERS:")

decrypt(dymessage, dyke) elif choies==3:

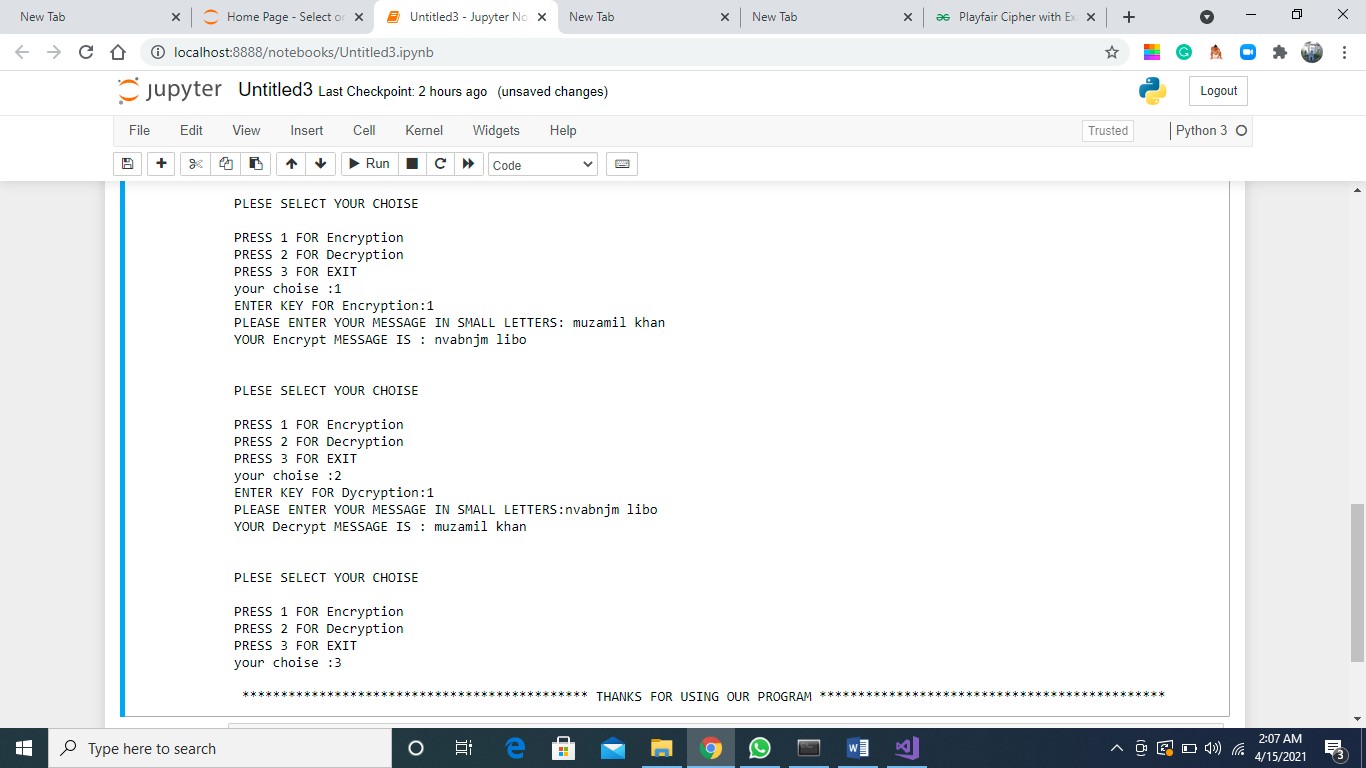
print("\n \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* THANKS

FOR USING OUR PROGRAM \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*") break

else:

print("sorry you select invalid choise. ")

# SCREENSHOT OF OUTPUT:



**…………………..THE END…………………..**