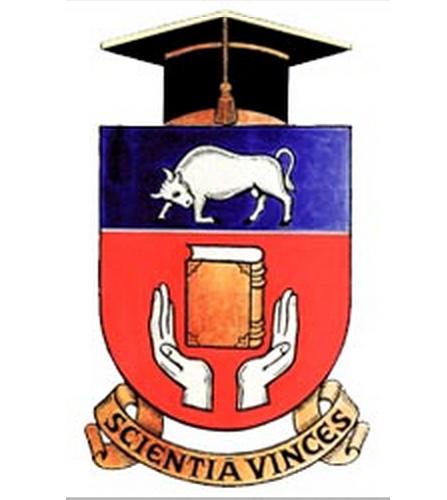
Universitatea de Stat din Tiraspol

Facultatea de Fizica Matematica si Informatica



Cifrul Vigenere

**Elaborat de**: Dordea Pavel, student gr.3i

**Verificat**: Globa Angela

**Chișinău, 2021**

**Sistemul Vigenere**

***#Despre***

**Cifrul Vigenère** este o metodă de criptare care folosește o serie de cifruri Cezar diferite bazate pe literele unui cuvânt-cheie. Este o formă simplă de substituție polialfabetică.

Cifrul Vigenère a fost reinventat de multe ori. Metoda a fost descrisă inițial de către Giovan Batista Belaso în 1553, în cartea sa *La cifra del. Sig. Giovan Batista Belaso*; totuși, schema a fost atribuită greșit mai târziu lui Blaise de Vigenère în secolul al XIX-lea, și este cunoscută acuma ca "cifrul Vigenère".

Acest cifru este cunoscut deoarece deși este ușor de înțeles și implementat, pare pentru începători imposibil de spart; acestui fapt i se datorează descrierea **le chiffre indéchiffrable** (franceză, "cifrul indescifrabil"). În consecință, mulți oameni au încercat să implementeze scheme de criptare care sunt, esențialmente, cifruri Vigenère, doar ca să fie sparte.

***#COD***

Criptare

def encrypt(message, key): #functia de criptare

crypted.delete(0,tk.END)

message = clear\_message.get().lower()

key = cheie.get().lower()

encrypted = ""

split\_message = [

message[i : i + len(key)] for i in range(0, len(message), len(key))

] # despartim mesajul in bucati de lungimea key-ului

for each\_split in split\_message:

i = 0

for letter in each\_split:

number = (letter\_to\_index[letter] + letter\_to\_index[key[i]]) % len(alphabet)

#convertim fiecare litera in numarul de oridne adunam key-a (mod26)

encrypted += index\_to\_letter[number]

#convertim din numerele de ordine in literele corespunzatoare

i += 1

L3["text"]="MESAJUL CRIPTAT ESTE :"

crypted.insert(0,encrypted)

crypted.grid(row=5,column=2,sticky='ne') #returnam mesajul criptat

Decriptare

def decrypt(cipher, key):

cipher = crypted.get().lower()

key =reverse\_key.get().lower()

decrypted = ""

split\_encrypted = [

cipher[i : i + len(key)] for i in range(0, len(cipher), len(key))

] # despartim mesajul in bucati de lungimea key-ului

for each\_split in split\_encrypted:

i = 0

for letter in each\_split:

number = (letter\_to\_index[letter] - letter\_to\_index[key[i]]) % len(alphabet)

#convertim fiecare litera in numarul de oridne si scadem key-a (mod26)

decrypted += index\_to\_letter[number]

#convertim din numerele de ordine in literele corespunzatoare

i += 1

L8.grid(row=12,column=0,sticky='w')

clr\_message.insert(0,decrypted)

clr\_message.grid(row=12,column=2,sticky='ne') #returnam mesajul criptat

**Corpul**

# -\*- coding: utf-8 -\*-

"""

Created on Tue Nov 16 21:10:52 2021

@author: Pavel

"""

import tkinter as tk

root= tk.Tk()

root.resizable(0,0)

root.title('Sistemul "Vigenere" by Pavel Dordea')

root.configure(bg='#164A41')

#\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_pentru user input\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

L1=tk.Label(root,text="Introdu mesajul clar :",bg='#164A41',fg='#FFFFFF',font='Helvetica 14 bold')

clear\_message = tk.Entry(root,bg='white',fg='black',font='Helvetica 13 bold')

L2=tk.Label(root,text="Introdu cheia secreta :",bg='#164A41',fg='#FFFFFF',font='Helvetica 14 bold')

cheie = tk.Entry(root,bg='white',fg='black',font='Helvetica 13 bold')

alphabet = "abcdefghijklmnopqrstuvwxyz "

letter\_to\_index = dict(zip(alphabet, range(len(alphabet))))#transformam din litere in numarul de ordine din alfabet

index\_to\_letter = dict(zip(range(len(alphabet)), alphabet))#transformam din numarul de ordine din alfabet in litere

message =''

key=''

def show\_user():

L1.grid(row=2,column=0,sticky="w")

L2.grid(row=3,column=0,sticky="w")

clear\_message.grid(row=2,column=2,sticky='ne')

cheie.grid(row=3,column=2,sticky='ne')

L3.grid(row=5,column=0,sticky='w')

button\_crypt.grid(row=4,columnspan=3,sticky='we')

L4.grid(row=6,columnspan=3,sticky='w')

button\_show.grid(row=7,column=2,sticky='ne')

b1.grid\_forget()

l3.grid\_forget()

b2.grid\_forget()

def show\_file():

L1.grid\_forget()

L2.grid\_forget()

clear\_message.grid\_forget()

cheie.grid\_forget()

L3.grid\_forget()

button\_crypt.grid\_forget()

L4.grid\_forget()

button\_show.grid\_forget()

b1.grid(row=2,column=0,sticky='w')

l3.grid(row=2,column=1)

b2.grid(row=2,column=2,sticky='ne')

crypted.grid\_forget()

L5.grid\_forget()

button\_hide.grid\_forget()

L6.grid\_forget()

crypted\_message.grid\_forget()

L7.grid\_forget()

reverse\_key.grid\_forget()

button\_decrypt.grid\_forget()

L8.grid\_forget()

clr\_message.grid\_forget()

def encrypt(message, key): #functia de criptare

crypted.delete(0,tk.END)

message = clear\_message.get().lower()

key = cheie.get().lower()

encrypted = ""

split\_message = [

message[i : i + len(key)] for i in range(0, len(message), len(key))

] # despartim mesajul in bucati de lungimea key-ului

for each\_split in split\_message:

i = 0

for letter in each\_split:

number = (letter\_to\_index[letter] + letter\_to\_index[key[i]]) % len(alphabet)

#convertim fiecare litera in numarul de oridne adunam key-a (mod26)

encrypted += index\_to\_letter[number]

#convertim din numerele de ordine in literele corespunzatoare

i += 1

L3["text"]="MESAJUL CRIPTAT ESTE :"

crypted.insert(0,encrypted)

crypted.grid(row=5,column=2,sticky='ne') #returnam mesajul criptat

def decrypt(cipher, key):

cipher = crypted.get().lower()

key =reverse\_key.get().lower()

decrypted = ""

split\_encrypted = [

cipher[i : i + len(key)] for i in range(0, len(cipher), len(key))

] # despartim mesajul in bucati de lungimea key-ului

for each\_split in split\_encrypted:

i = 0

for letter in each\_split:

number = (letter\_to\_index[letter] - letter\_to\_index[key[i]]) % len(alphabet)

#convertim fiecare litera in numarul de oridne si scadem key-a (mod26)

decrypted += index\_to\_letter[number]

#convertim din numerele de ordine in literele corespunzatoare

i += 1

L8.grid(row=12,column=0,sticky='w')

clr\_message.insert(0,decrypted)

clr\_message.grid(row=12,column=2,sticky='ne') #returnam mesajul criptat

def show():

button\_show.grid\_forget()

L5.grid(row=8,columnspan=3,sticky='w')

button\_hide.grid(row=7,column=0,sticky='w')

L6.grid(row=9,column=0,sticky='w')

crypted\_message.grid(row=9,column=2,sticky='ne')

L7.grid(row=10,column=0,sticky='w')

reverse\_key.grid(row=10,column=2,sticky='ne')

button\_decrypt.grid(row=11,columnspan=3,sticky='we')

def hide():

L5.grid\_forget()

button\_show.grid(row=7,column=2,sticky='ne')

button\_hide.grid\_forget()

L6.grid\_forget()

crypted\_message.grid\_forget()

L7.grid\_forget()

reverse\_key.grid\_forget()

button\_decrypt.grid\_forget()

L8.grid\_forget()

clr\_message.grid\_forget()

#\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

def show\_crypt():

key\_crypt.delete(0,tk.END)

L1['text']= 'Introdu cheia de criptare:'

L1.grid(row=3,column=0,sticky="w")

key\_crypt.grid(row=3,column=2,sticky='ne')

butt\_crypt.grid(row=4,columnspan=3,sticky='we')

l3["text"]=" "

def show\_decrypt():

key\_crypt.delete(0,tk.END)

L1['text']= 'Introdu cheia de decriptare:'

L1.grid(row=3,column=0,sticky="w")

key\_crypt.grid(row=3,column=2,sticky='ne')

butt\_decrypt.grid(row=4,columnspan=3,sticky='we')

l3["text"]=" "

def getmsg():

message = open('Message.txt', 'r+').read().lower().replace('\n',' ')

return message

def getkey():

cheie = key\_crypt.get()

return cheie

def fencrypt(): #functia de criptare

message=getmsg()

key =getkey()

encrypted = ""

split\_message = [

message[i : i + len(key)] for i in range(0, len(message), len(key))

] # despartim mesajul in bucati de lungimea key-ului

for each\_split in split\_message:

i = 0

for letter in each\_split:

number = (letter\_to\_index[letter] + letter\_to\_index[key[i]]) % len(alphabet)

#convertim fiecare litera in numarul de oridne adunam key-a (mod26)

encrypted += index\_to\_letter[number]

#convertim din numerele de ordine in literele corespunzatoare

i += 1

f = open("Encrypted.txt", "w+")

f.write(encrypted)#returnam mesajul criptat

l3["text"]="Success ! "

def getenc():

encrypted = open('Encrypted.txt', 'r+').read().lower().replace('\n',' ')

return encrypted

def fdecrypt():

cipher = getenc()

key =getkey()

decrypted = ""

split\_encrypted = [

cipher[i : i + len(key)] for i in range(0, len(cipher), len(key))

] # despartim mesajul in bucati de lungimea key-ului

for each\_split in split\_encrypted:

i = 0

for letter in each\_split:

number = (letter\_to\_index[letter] - letter\_to\_index[key[i]]) % len(alphabet)

#convertim fiecare litera in numarul de oridne si scadem key-a (mod26)

decrypted += index\_to\_letter[number]

#convertim din numerele de ordine in literele corespunzatoare

i += 1

f = open("Decrypted.txt", "w+")

f.write(decrypted)

l3["text"]="Success ! "

#\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

#\_\_\_\_\_\_\_\_\_\_\_\_\_\_pentru user input\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

user = tk.Button(root,text="De la user",command=show\_user ,bg='#4D774E',fg='#FFFFFF',font='Helvetica 12 bold')

user.grid(row=1,columnspan=3,padx=200,pady=10,sticky='we')

button\_crypt = tk.Button(root,text="Cripteaza",command=lambda: encrypt(message, key) ,bg='#4D774E',fg='#FFFFFF',font='Helvetica 12 bold')

L3=tk.Label(root,text="",bg='#164A41',fg='red',font='Helvetica 12 bold')

crypted=tk.Entry(root,bg='white',fg='black',font='Helvetica 13 bold')

L4=tk.Label(root,text="\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_",bg='#164A41',fg='#ffffff',font='Helvetica 12 bold')

button\_show = tk.Button(root,text="Mai mult",command=show ,bg='#4D774E',fg='#FFFFFF',font='Helvetica 7 bold')

button\_hide = tk.Button(root,text="Mai putin",command=hide ,bg='#4D774E',fg='#FFFFFF',font='Helvetica 7 bold')

L5=tk.Label(root,text="\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_",bg='#164A41',fg='#ffffff',font='Helvetica 12 bold')

L6=tk.Label(root,text="Introdu mesajul criptat :",bg='#164A41',fg='#ffffff',font='Helvetica 12 bold')

crypted\_message=tk.Entry(root,bg='white',fg='black',font='Helvetica 13 bold')

L7=tk.Label(root,text="Introdu cheia secreta :",bg='#164A41',fg='#ffffff',font='Helvetica 12 bold')

reverse\_key=tk.Entry(root,bg='white',fg='black',font='Helvetica 13 bold')

button\_decrypt=tk.Button(root,text="Decripteaza",command=lambda: decrypt(message, key) ,bg='#4D774E',fg='#FFFFFF',font='Helvetica 12 bold')

L8=tk.Label(root,text="MESAJUL CLAR ESTE :",bg='#164A41',fg='red',font='Helvetica 12 bold')

clr\_message = tk.Entry(root,bg='white',fg='black',font='Helvetica 13 bold')

#\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_pentru fisiere\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

fisiere = tk.Button(root,text="Din fisier",command=show\_file ,bg='#4D774E',fg='#FFFFFF',font='Helvetica 12 bold')

fisiere.grid(row=0,columnspan=3,padx=200,pady=10,sticky='we')

Lf1=tk.Label(root,text="",bg='#164A41',fg='#FFFFFF',font='Helvetica 14 bold')

key\_crypt = tk.Entry(root,bg='white',fg='black',font='Helvetica 13 bold')

b1 = tk.Button(root,text="Codificare",command= show\_crypt ,bg='#4D774E',fg='#FFFFFF',font='Helvetica 12 bold')

l3=tk.Label(root,text=" ",bg='#164A41',fg='darkred',font='Helvetica 12 bold')

b2 = tk.Button(root,text="Decodificare",command= show\_decrypt ,bg='#4D774E',fg='#FFFFFF',font='Helvetica 12 bold')

butt\_crypt = tk.Button(root,text="Cripteaza",command= fencrypt ,bg='#4D774E',fg='#FFFFFF',font='Helvetica 12 bold')

butt\_decrypt = tk.Button(root,text="Decripteaza",command= fdecrypt ,bg='#4D774E',fg='#FFFFFF',font='Helvetica 12 bold')

root.mainloop()

***#Screenshots***

