

Computer Security Project.

Presented to: Prof. Dr. Mahmoud E. Elshishtawy.

Dr. Nagham Yahya.

Prepared by: Ahmed Abdel Moneim Abdel Halim.

Group: G1.

ID: 91271.

Computer Security Techniques:

- 1) Caesar Cipher Technique.
- 2) PolyAlphabetic Cipher Technique.
- 3) Transposition Technique.

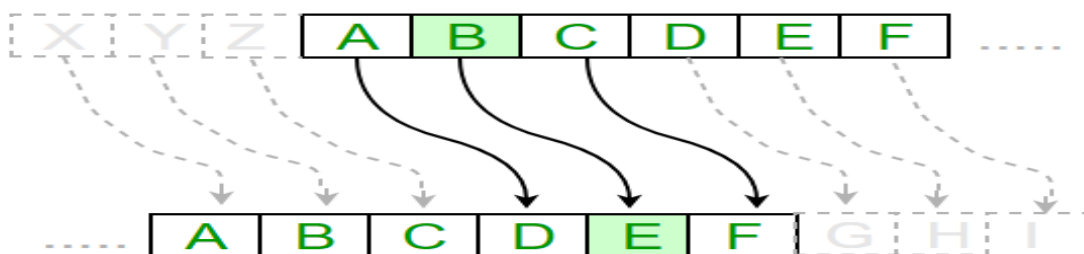
In cryptography, a cipher is an algorithm for performing encryption or decryption a series of well-defined steps that can be followed as a procedure. An alternative , less common term is encipherment.

To encipher or encode is to convert information into cipher or code.

1) Caesar cipher:

In cryptography, a Caesar cipher, also Known as Caesar's cipher, the shift cipher , Caesar's code or Caesar shift, is one of the simplest and most widely known encryption techniques.

The action of a Caesar cipher is to replace each plaintext letter with a different one a fixed number of places down the alphabet. The cipher uses a left shift of three, so that each occurrence of E in the plaintext becomes B in the ciphertext.



2) PolyAlphabetic cipher:

A polyalphabetic cipher substitution, using multiple substitutions alphabets. The Vigenère cipher is probably the best-known example of a polyalphabetic cipher, though it is a simplified special case. The Enigma machine is more complex but is still fundamentally a polyalphabetic substitution cipher.

The Rule: $C_i = (\text{Key-Letter} + \text{Message-Letter}) \bmod 26$.

Plaintext	A	T	T	A	C	K	A	T	D	A	W	N
Key	L	E	M	O	N	L	E	M	O	N	L	E
Shift	+11	+4	+12	+14	+13	+11	+4	+12	+14	+13	+11	+4
Ciphertext	L	X	F	O	P	V	E	F	R	N	H	R

3) Transposition cipher:

It is a method of encryption which scrambles the positions of characters (transposition) without changing the character themselves. Is achieved by performing some sort of permutation on the plaintext letters.

Transposition Cipher

4	3	1	5	2
N	A	T	S	R
I	S	P	T	O
X	N	I	X	O

plain text: TRANSPOSITION

keyword: 43152

ciphertext: NIXASNTPISTXPOO

Code Of The Project:

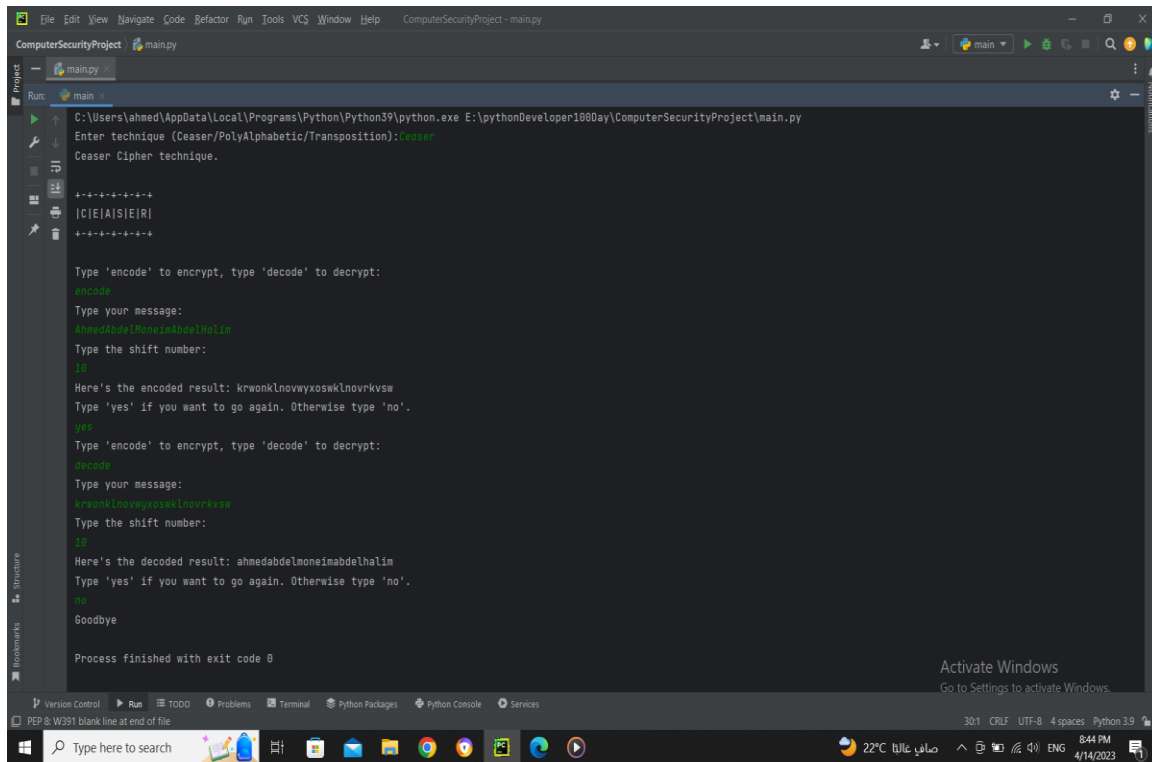
```
File Edit View Navigate Code Refactor Run Tools VCS Window Help ComputerSecurityProject - main.py
ComputerSecurityProject main.py
1 #BY-Ahmed Abdel Monem Abdel Halim
2 from art import logo, logo2, logo3
3 #print(logo)
4 Enter_technique = str(input("Enter technique (Caesar/PolyAlphabetic/Transposition):").lower())
5 if Enter_technique == 'caesar':
6     print("Caesar Cipher technique.")
7     print(logo1)
8     alphabet = ['a', 'b', 'c', 'd', 'e', 'f', 'g', 'h', 'i', 'j', 'k', 'l', 'm',
9                 'n', 'o', 'p', 'q', 'r', 's', 't', 'u', 'v', 'w', 'x', 'y', 'z',
10                'a', 'b', 'c', 'd', 'e', 'f', 'g', 'h', 'i', 'j', 'k', 'l', 'm',
11                'n', 'o', 'p', 'q', 'r', 's', 't', 'u', 'v', 'w', 'x', 'y', 'z']
12
13 def caesar(start_text, shift_amount, cipher_direction):
14     end_text = ""
15     if cipher_direction == "decode":
16         shift_amount *= -1
17     for char in start_text:
18         if char in alphabet:
19             position = alphabet.index(char)
20             new_position = position + shift_amount
21             end_text += alphabet[new_position]
22         else:
23             end_text += char
24     print(f"Here's the {cipher_direction}d result: {end_text}")
25 should_end = False
26 while not should_end:
27     direction = input("Type 'encode' to encrypt, type 'decode' to decrypt:\n")
28     text = input("Type your message:\n").lower()
29     shift = int(input("Type the shift number:\n"))
30     shift = shift % 26
31     caesar(start_text=text, shift_amount=shift, cipher_direction=direction)
32     restart = input("Type 'yes' if you want to go again. Otherwise type 'no'.")
33     if restart == "no":
34         should_end = True
35         print("Goodbye")
36 elif Enter_technique == 'polyalphabetic':
37     print("PolyAlphabetic cipher technique.")
38     print(logo2)
39     #index from 0 to 25
40     Alphabet = ['a', 'b', 'c', 'd', 'e', 'f', 'g', 'h', 'i', 'j', 'k', 'l', 'm',
41                'n', 'o', 'p', 'q', 'r', 's', 't', 'u', 'v', 'w', 'x', 'y', 'z']
42     message_Text = str(input("Enter Message Text:"))
43     message_Key = str(input("Enter Key:"))
44     cipher_Text = ""
45     #Convert Message to List
46     Message_Text_List = list(message_Text.strip(" "))
47     Message_Key_List = list(message_Key.strip(" "))
48     #print(Message_Text_List)
49     #print(Message_Key_List)
50     #print(Message_Key_List)
51     #print(Message_Key_List)
52     for num1 in range(0, len(Message_Text_List)):
53         if num1 >= len(Message_Key_List):
54             Message_Key_List = Message_Key_List
55             letter_Text_index = Alphabet.index(Message_Text_List[num1].lower())
56             letter_Key_index = Alphabet.index(Message_Key_List[num1].lower())
57             cipher_index = (int(letter_Text_index) + int(letter_Key_index)) % 26
58             cipher_Text += Alphabet[cipher_index]
59         print("The Cipher Text: ", cipher_Text)
60     #print(cipher_Text)
61 elif Enter_technique == 'transposition':
62     print("Transposition cipher technique.")
```

```
File Edit View Navigate Code Refactor Run Tools VCS Window Help ComputerSecurityProject - main.py
ComputerSecurityProject main.py
30 caesar(start_text=text, shift_amount=shift, cipher_direction=direction)
31 restart = input("Type 'yes' if you want to go again. Otherwise type 'no'.")
32 if restart == "no":
33     should_end = True
34     print("Goodbye")
35 elif Enter_technique == 'polyalphabetic':
36     print("PolyAlphabetic cipher technique.")
37     print(logo2)
38     #index from 0 to 25
39     Alphabet = ['a', 'b', 'c', 'd', 'e', 'f', 'g', 'h', 'i', 'j', 'k', 'l', 'm',
40                'n', 'o', 'p', 'q', 'r', 's', 't', 'u', 'v', 'w', 'x', 'y', 'z']
41     message_Text = str(input("Enter Message Text:"))
42     message_Key = str(input("Enter Key:"))
43     cipher_Text = ""
44     #Convert Message to List
45     Message_Text_List = list(message_Text.strip(" "))
46     Message_Key_List = list(message_Key.strip(" "))
47     #print(Message_Text_List)
48     #print(Message_Key_List)
49     #print(Message_Key_List)
50     #print(Message_Key_List)
51     for num1 in range(0, len(Message_Text_List)):
52         if num1 >= len(Message_Key_List):
53             Message_Key_List = Message_Key_List
54             letter_Text_index = Alphabet.index(Message_Text_List[num1].lower())
55             letter_Key_index = Alphabet.index(Message_Key_List[num1].lower())
56             cipher_index = (int(letter_Text_index) + int(letter_Key_index)) % 26
57             cipher_Text += Alphabet[cipher_index]
58         print("The Cipher Text: ", cipher_Text)
59     #print(cipher_Text)
60 elif Enter_technique == 'transposition':
61     print("Transposition cipher technique.")
```

```
ComputerSecurityProject - main.py
60 elif Enter_technique == 'transposition':
61     print("Transposition Cipher technique.")
62     print(logos)
63     #Message:"depositatenmillionpoundintbrahiemaccounttyz"#42.
64     #Key:"2641753"7.
65     message_Text = str(input("Enter Message Text:"))
66     message_Key = str(input("Enter Key:"))
67     cipher_Text = ""
68     if len(message_Text)==42 and len(message_Key)==7:
69         Message_Text_List = list(message_Text.strip(" "))
70         Message_Key_List = list(message_Key.strip(" "))
71         Column=[]
72         Number = 0
73         len_Message = len(Message_Key_List)
74         for run in range(0,len(Message_Key_List)):
75             Column.append(Message_Text_List[0+Number])
76             Column.append(Message_Text_List[len_Message+Number])
77             Column.append(Message_Text_List[len_Message+len_Message+Number])
78             Column.append(Message_Text_List[len_Message+len_Message+len_Message+Number])
79             Column.append(Message_Text_List[len_Message+len_Message+len_Message+len_Message+Number])
80             Column.append(Message_Text_List[len_Message+len_Message+len_Message+len_Message+Number])
81             Number+=1
82         print(len(Message_Text_List))
83         Value = {
84             f"{Message_Key_List[0]}": Column[0:0],
85             f"{Message_Key_List[1]}": Column[0:12],
86             f"{Message_Key_List[2]}": Column[12:18],
87             f"{Message_Key_List[3]}": Column[18:24],
88             f"{Message_Key_List[4]}": Column[24:30],
89             f"{Message_Key_List[5]}": Column[30:36],
90             f"{Message_Key_List[6]}": Column[36:42]}
91         sorted_value = sorted(Value)
```

```
ComputerSecurityProject - main.py
75 Message_Key_List = list(message_Key.strip(" "))
76 Column=[]
77 Number = 0
78 len_Message = len(Message_Key_List)
79 for run in range(0,len(Message_Key_List)):
80     Column.append(Message_Text_List[0+Number])
81     Column.append(Message_Text_List[len_Message+Number])
82     Column.append(Message_Text_List[len_Message+len_Message+Number])
83     Column.append(Message_Text_List[len_Message+len_Message+len_Message+Number])
84     Column.append(Message_Text_List[len_Message+len_Message+len_Message+len_Message+Number])
85     Column.append(Message_Text_List[len_Message+len_Message+len_Message+len_Message+Number])
86     Number+=1
87     print(len(Message_Text_List))
88     Value = {
89         f"{Message_Key_List[0]}": Column[0:0],
90         f"{Message_Key_List[1]}": Column[0:12],
91         f"{Message_Key_List[2]}": Column[12:18],
92         f"{Message_Key_List[3]}": Column[18:24],
93         f"{Message_Key_List[4]}": Column[24:30],
94         f"{Message_Key_List[5]}": Column[30:36],
95         f"{Message_Key_List[6]}": Column[36:42]}
96     sorted_value = sorted(Value)
97     cipher_Text = Value[sorted_value[0]]+Value[sorted_value[1]]+Value[sorted_value[2]]+Value[sorted_value[3]]
98     +Value[sorted_value[4]]+Value[sorted_value[5]]+Value[sorted_value[6]]
99     cipher_Text = "".join(cipher_Text)
100     print(cipher_Text)
101 else:
102     print("Please Enter 42 Message Letter and 7 Key Number.")
103 else:
104     print("Wrong Input Try Agen.")
105     =====
106     =====
```

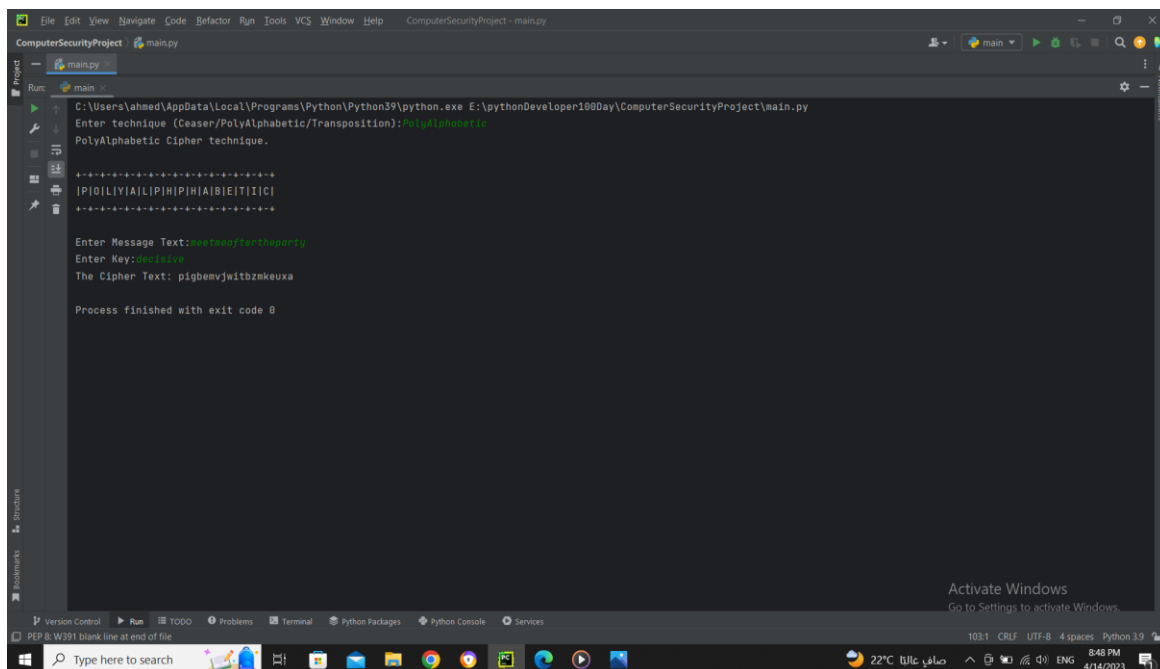
The Output of Program:



```
ComputerSecurityProject - main.py
C:\Users\ahmed\AppData\Local\Programs\Python\Python39\python.exe E:\pythonDeveloper180day\ComputerSecurityProject\main.py
Enter technique (Ceaser/PolyAlphabetic/Transposition):Caesar
Ceaser Cipher technique.

+++++
|C|E|A|S|E|R|
+++++

Type 'encode' to encrypt, type 'decode' to decrypt:
encode
Type your message:
ahmedabdelmoneimabdelhalim
Type the shift number:
10
Here's the encoded result: krwonklnovwvxoswklnovrkvsw
Type 'yes' if you want to go again. Otherwise type 'no'.
yes
Type 'encode' to encrypt, type 'decode' to decrypt:
decode
Type your message:
krwonklnovwvxoswklnovrkvsw
Type the shift number:
10
Here's the decoded result: ahmedabdelmoneimabdelhalim
Type 'yes' if you want to go again. Otherwise type 'no'.
no
Goodbye
Process finished with exit code 0
```



```
ComputerSecurityProject - main.py
C:\Users\ahmed\AppData\Local\Programs\Python\Python39\python.exe E:\pythonDeveloper180day\ComputerSecurityProject\main.py
Enter technique (Ceaser/PolyAlphabetic/Transposition):PolyAlphabetic
PolyAlphabetic Cipher technique.

+++++
|P|O|L|Y|A|L|P|H|P|H|A|B|E|T|I|C|
+++++

Enter Message Text:asafwasafthaparty
Enter Key:asafwasaf
The Cipher Text: pibemvjwibzmkexa

Process finished with exit code 0
```

The screenshot shows a Python IDE window titled "ComputerSecurityProject - main.py". The code in the editor defines a transposition cipher function. The output in the console shows the program running successfully, with the input message "repositsteneallianpawndinibnehieaccountry" being encrypted using the key "42" to produce the ciphertext "onnendalnactlurczpeoiuiiobayetidhosmpint".

```
File Edit View Navigate Code Refactor Run Tools VCS Window Help ComputerSecurityProject - main.py
ComputerSecurityProject main.py
main.py
Run: main
C:\Users\ahmed\AppData\Local\Programs\Python\Python39\python.exe E:\pythonDeveloper100Day\ComputerSecurityProject\main.py
Enter technique (Ceaser/PolyAlphabetic/Transposition): Transposition
Transposition Cipher technique.
+-----+
|T|R|A|N|S|P|O|S|I|T|I|O|N|
+-----+

Enter Message Text: repositsteneallianpawndinibnehieaccountry
Enter Key: 42
42
onnendalnactlurczpeoiuiiobayetidhosmpint

Process finished with exit code 0
```

#####END-END#####

#####