

**Samsung Innovation Campus**

Cryptography: Cipher

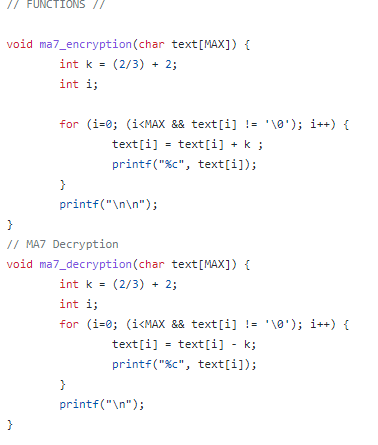
[encryption and decryption]

Team 2:

Eyad Elfar

Youssef Hisham

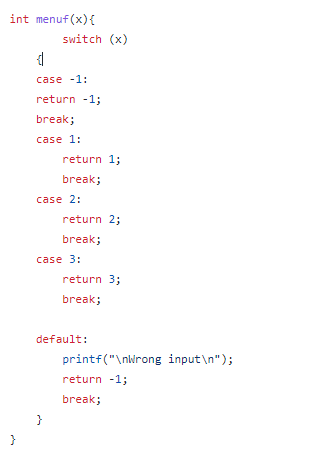
Mahmoud Adel



**In This code:**

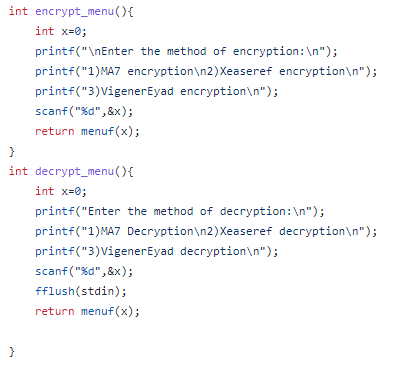
**This is MA7 encryption and decryption function.**

1. **We made a key that we use to add or subtract.**
2. **We made a loop that selects every letter individually at the word that user input. And add the key to it and print the letter. We do this process to every letter.**
3. **In decryption like encryption but we subtract the key from the letter.**

****

**In This Code :**

**We use this function to take input from the user and return the right process that he needs.**

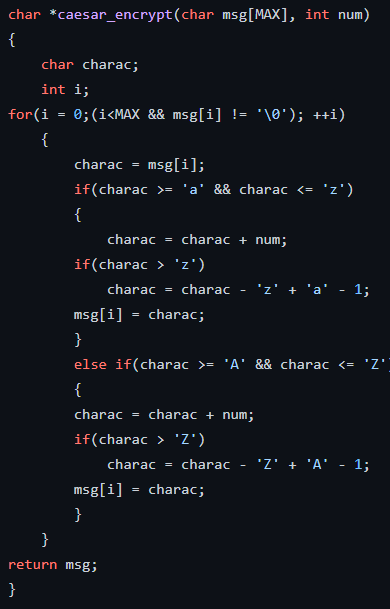
****

**In This Code :**

**We show the user which process he needs. This is the menu Which facilitates the program for the user.**

**It makes you choose between encryption with all kinds of it and decryption with all kinds of it.**

**And take the answer from the user (input).then , call the menuf function to do the process which depends on the answer of the user.**

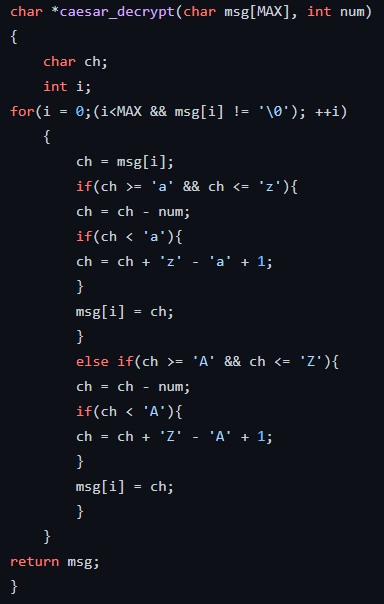


In this Code, the 2nd Encryption option is Caesar Cipher   
One of the simplest encryption styles basically taking a string from the user and taking a key.  
The key is the secret communication between two sides to encrypt and decrypt the message.

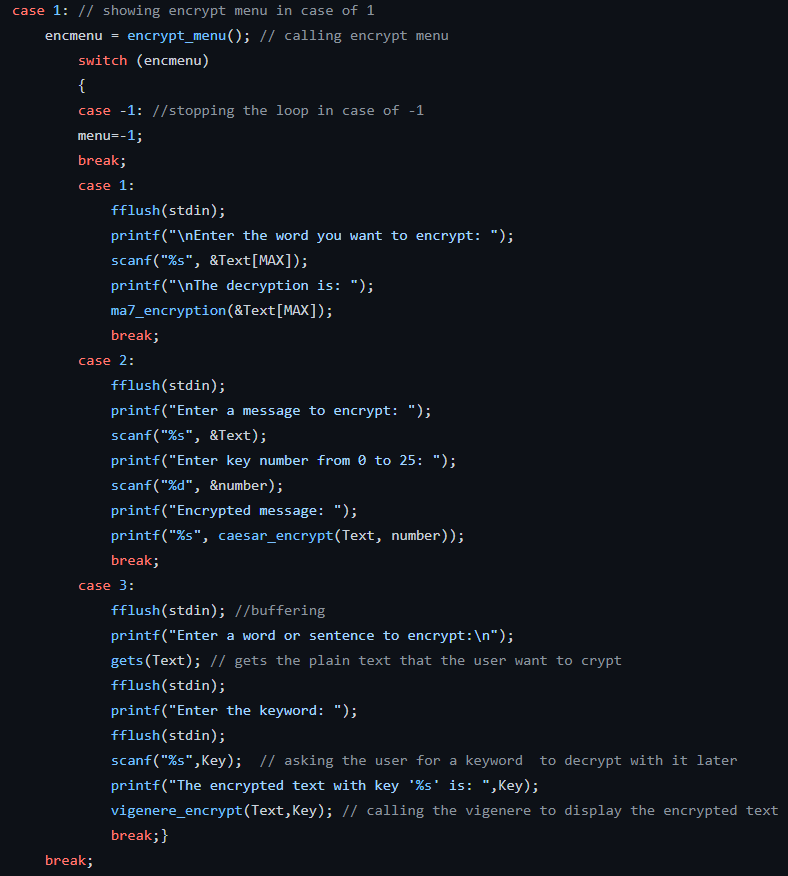
The Key is simply the number that will shift every character in a string to it’s ASCII code one

For example

If a character is equal to 50 for example and we chose the key to be 3   
Then this character will be 50 + 3 = 53 so it will replace the character(50) with the character(53)



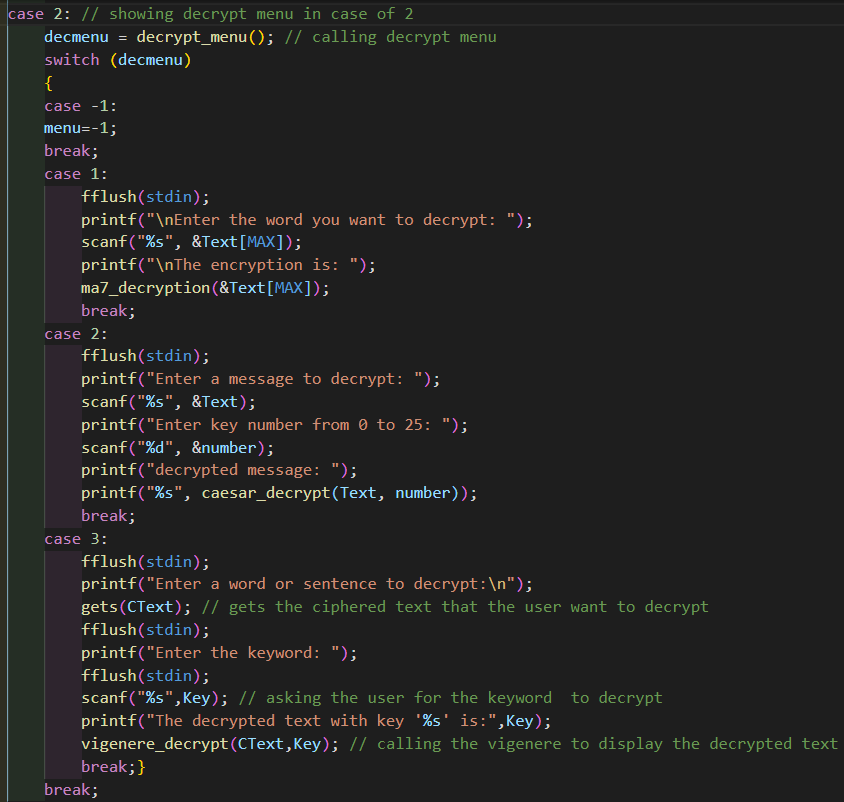
The Caesar decryption is the same as the encryption but this time we just make a minus in the characters ascii instead of plus and shifting   
So basically we shift backward not forward   
So you have to know the “Key” as to decrypt the message right. This is the point of encrypting messages tho.

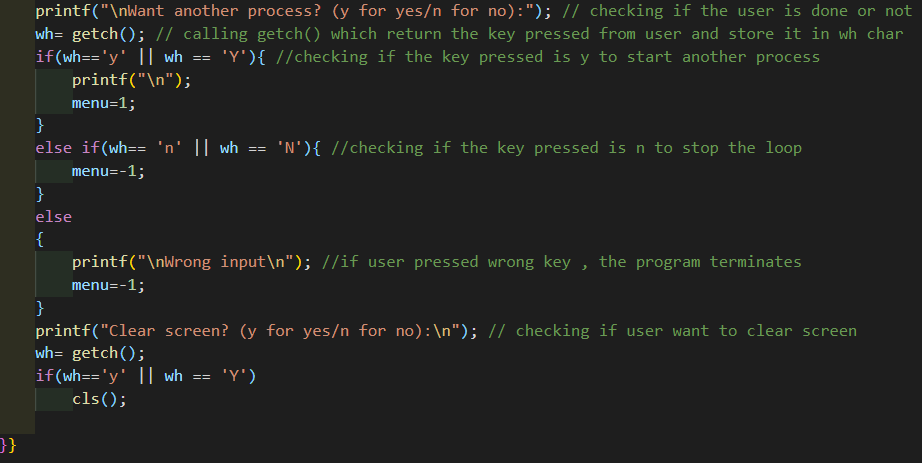


This part in case if the user chose (1) to encrypt menu.

So it will give him the option to enter way of encryption from the 3 types   
Then entering the word/string to be encrypted and a key if it is required in the encryption type.

So we have 3 cases (3 types of encryption)  
And the required inputs based on the chosen case.

In this case , the user want to decrypt some ciphered text, so we will check what ciphered text he entered to decrypt with it , after selecting the case of decryption we call the function whether it’s first or second or vigenerEyad, to get input from user and decrypt it with the called function.

Here at the end of our while loop, we check if the user want another process or he is done by using getch() which return the key pressed if it is y so we continue the loop if its not we make menu = -1 which mean the loop will stop, if the user want another process we ask him if he want to work with same screen or clear the screen(it is fake clear screen which add many new lines so the screen appear cleared but if user scrolled up he can find old process and that is because user may forget his old entry or something).

VigenérEyad Cipher technique:

**What is VigenerEyad cipher?**

VigenerEyad cipher is a cipher just like vigenere that can be decrypted with any site decrypting vigenere but what is the secret ?, that if the plaintext contains spaces only the word before the space can be encrypted with any site but the word after space can’t be decrypted correctly except with vigenerEyad decrypt, why because we are special , normal vigenere ignores special characters we don’t , we ignore it apparently but in depth the key counts as it is.

**how do it works ?**

You write the plain text , the key and it will shift it right when encrypting , shift it left when decrypting with the key value.

i.e. you entered “eyad” to be encrypted, and entered the key “xd” ,

here is how the process is done

“eyad”+”xd” , it will take the first character of text and adding the first character with key

((‘e’ - 97+’x’-97) % number of alphabets) + 97 (first character of alphabets )

= ((101-97 (in ascii) + 120-97) % 26) + 97 = (4 + 23) %26 + 97 =(27%26)+97 = 98= ‘b’ from ascii ,‘y’ will be with d

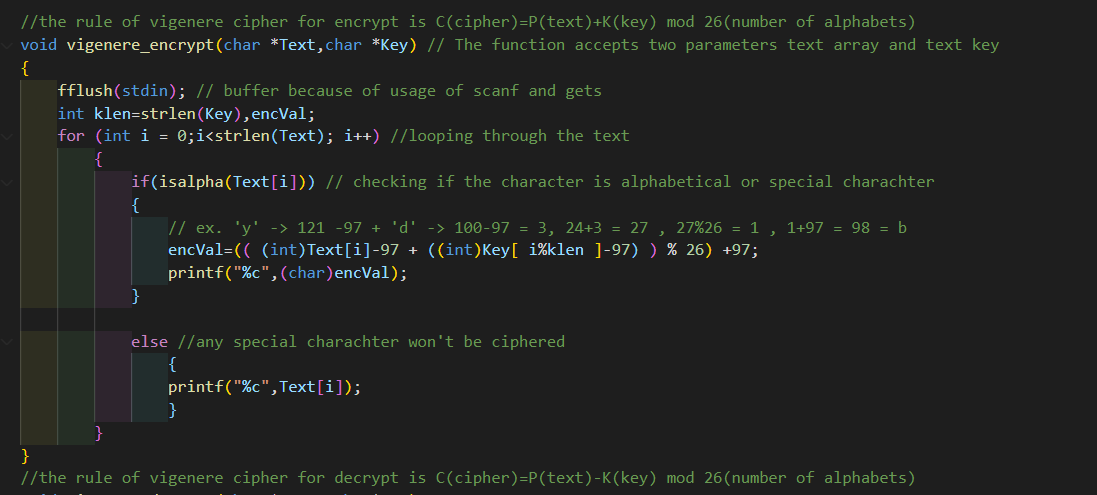
And ‘a’ will be with ‘x’ again because the key is modded to its length so each time it is at the last value it resets from ( 0 ) so ‘a’ is with ‘x’ and d is with ‘d’ so “eyad” with “xd” will be “bbxg”

At decrypting:

We just invert the sign from ‘+’ to ‘-’ so let's decrypt “bbxg” with key “xd”,

((‘b’ - 97-’x’-97 )% number of alphabets ) + 97

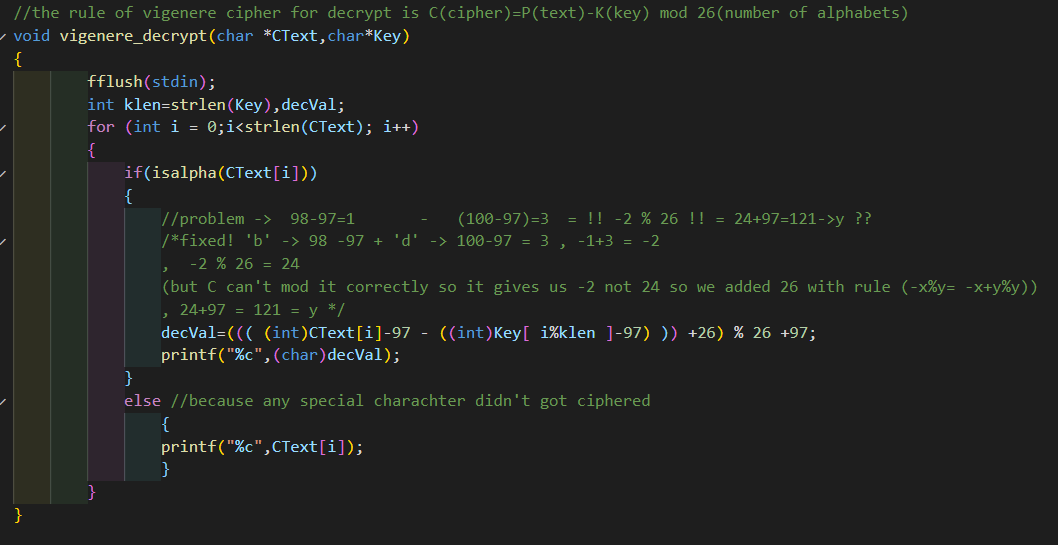
So (98-97 - 120-97)%26 + 97 = (-22%26)+97= 4 + 97 = 101 which is ‘e’ after decrypting you will get “eyad” back.



In this code, we implemented vigenere encryption with the formula Cipher=P(plain text)+K[keyword text] % n , where n is the number of alphabets we use,

We looping through the plain text checking if the character is alphabetic or not if its alphabetic we shift right it(adding) by the value of key, we made in the condition Key[i%klen] because we need the key to be repeated if its shorter that the plaintext.

If the character isn’t alpha we print it as it is because its invalid input to encrypt in vigenere.



In this code, we loop through the ciphered text and do just the inverse of encryption we shift the key left from the ciphered text so we get our plain text back easily.

**Work done by:**

Eyad Elfar

Mahmoud Adel

Youssef Hisham

**Facilitators:**

Ali Hammed

Nada Hussien