REPORT:

Firstly all the variables have been defined and assigned to specific values.

**Functions have been done for :**

* Conversion from string to binary using password as parameter. Extracts characters one by one to make conversion and after each word, after detecting 8 bits in sequence, the function inserts a space for the next 8 bits of the next word to be converted. After every conversion, the function pushes the calculated value which will be 1 or 0 to the left using ‘<<’. In the end, patches of 8 bits are obtained for each word.
* Encryption of cesar using text as parameter. For both uppercase and lowercase alphabets. Extraction of character one by one and adding the key that is between 1 and 9 which is the “decalage” to it to obtain another alphabet to be used for the encryption. If the key is 3 and chosen alphabet is ‘A’ then key + (ascii of alphabet) will give us in this case 3 + 65 =68, which is the letter C. The values of key and text have been returned to be used in the main program. To avoid special characters to be formed, I used for example the method: if(ch > 'z') { ch = ch + 'z' - 'a' + 1 which means, if character’s ascii obtained when encryption is done is greater than ascii of z, the ascii of the character will be added to ascii of z then subtracted to ascii of a then adding one to it will make us obtain a valid ascii code which will be between range. This applies to both uppercase and lowercase characters.

* Decryption of cesar using text as parameter. For both uppercase and lowercase alphabets. Extraction of character one by one and subtracting the key that is between 1 and 9 which is the “decalage” to it to obtain another alphabet to be used for the decryption. The alphabet obtained will be the same alphabet before encryption of data. For example if letter to be decrypted is C and key is 3 which is the “decalage”, results will give us A as ascii of character C which is 68 minus 3 gives us ascii value 65 which is A. To avoid special characters to be formed, I used for example the method: if(ch > 'z') { ch = ch + 'z' - 'a' + 1 which means, if character’s ascii obtained when encryption is done is greater than ascii of z, the ascii of the character will be added to ascii of z then subtracted to ascii of a then adding one to it will make us obtain a valid ascii code which will be between range. This applies to both uppercase and lowercase characters.

Then I opened a text file by doing : FILE\* fichier; fichier = fopen("TP1.txt","w"); , meaning to create a text file called TP1.txt as w which TP1.txt. Information about encryption will be inserted in it.

Afterwards, i asked the user to input a password in string which is later converted to binary using the conversion function.

Using the password, i found its length using the function strlength() and until the length isn’t the same as the incrementing value of i, it does the first XOR which is hash= hash ^ pwd[i] meaning that for every character of pwd, it does the XOR and keeps it in the variable hash.

For the second hash, i used k= hash ^ key in which hash is the XOR of the characters of password, key is the “decalage” and k is the variable storing the second XOR.

To do the final XOR of cesar and k, i used:

{ int atoi(const char \*str } which is the syntax to declare the function atoi() where it converts a string character to an integer value. As without it XOR cannot be performed on 2 different variable types.

{ xor= atoi(text) ^ k } This is the 3rd XOR of text and k and result obtained is being stored in xor variable.

Also, I declared a variable called choice where it will ask a user to choose between Encrypt or Decrypt on the first try.

To get accurate results, I used a function called strcmp() which is the string compare function to compare value of user input to the string “encrypt” or “decrypt”.

If choice is equals to encrypt, it will run the function encrypt() where it will encrypt the text by cesar, ouput the encrypted text, output the hash (first xor), output the second xor results, and the encrypted XOR text. In addition, it will also print the value of the XOR in the text file.

If the choice decrypt is selected on the first try, it will display: ‘ Enter text first in encrypt mode ! ‘

On the second choice command, it will ask a user to choose between Encrypt, Decrypt or Quit.

If the choice encrypt is selected on the second try, it will display: ‘ Already encrypted ! ‘

If the choice decrypt is selected on the second try, it will be as follows:

Firstly, runs the function decrypt() with text as parameter. Prints the decrypted message in the text file.

Does the XOR of k1 = xor ^ atoi(text) ; where the variable xor XOR text is being done with the function atoi() also here because of the different data types of each variable. This xor is done to revert back to K which is the second XOR result.

Then, XOR of k and key is done to find the hash which is the first XOR result. Code is as follows: hash = k ^ key; Where k is the second XOR result, Key is the “decalage” and hash is the variable to store the new XOR result.

Using the password again, i found its length using the function strlength() and until the length isn’t the same as the incrementing value of i, it does the third XOR which is hash= hash ^ pwd[i] meaning that for every character of pwd, it does the XOR and keeps it in the variable hash.

With all the three XOR now completed, original text can be obtained by removing the cesar encryption between them.

The new value of the hash and decrypted text are displayed.

The choice ‘ quit ‘ on the second try terminates the program by displaying: “ Program finished ! ”

Finally, the text document can be closed by using : “ fclose(fichier) “

The text file TP1.txt output contents as follows:

For example, after one run of the program entirely, the text file will have informations like:

Sentence: khushveer has an e30

Cesar encrypted : nkxvkyhhu kdv dq h30

XOR is : …ð$@…ð$@

Decrypted Cesar : khushveer has an e30

--------------------------------------------------------------------------------------------------------------------------------------------------------

Thank you.

Name: Khushveer Sujeebun

BTS SIO1