Mohammad Sufyan Azam, 2020312
Aamleen Ahmed, 2020002

NSC Assignment 2

We had to implement Project 0: the DES algorithm from scratch.

DES Implementation -

1. Constant Tables -

1. For Encryption and Decryption Purposes:
    a. Initial Permutation Table
    b. Inverse (or Final) Permutation Table
    c. Expansion Permutation *(for expanding the input block from 32 to 48 bits)*
    d. P-box Permutation
    e. 8 S-boxes Permutation *(for shrinking the input block from 48 to 32 bits)*

2. For the key:
    a. Permutation Choice 1 (PC1) *for shrinking the key from 64 to 56 bits*
    b. Permutation Choice 2 (PC2) *for shrinking the key further from 56 to 48 bits*

Note: These tables were taken from the slides and the book referenced in the slides.

2. Helper Functions -

There were six helper functions that were used in the program. These were:

1. string_to_binary(): *Converts a string to its binary values*
2. binary_to_string(): *Converts a sequence of binary bits into a string*
3. string_to_hex(): *Converts a string to its hexadecimal value*
4. hex_to_string(): *Converts a sequence of hexadecimal values to a string*
5. xor(): *Takes two binary input blocks and returns the xor of them*
6. permute(): *Takes the input block which needs to be permuted along with the permutation table and applies the table on the input block*

Mohammad Sufyan Azam, 2020312
Aamleen Ahmed, 2020002

3. DES Encryption -

```
def des_encryption(input_block, key, verifyRound = 0):
    bin_key = string_to_binary(key)
    bin_input_block = string_to_binary(input_block)
    keys = generate_keys(bin_key)
    plaintext = initial_permutation(bin_input_block)

    L, R = plaintext[:32], plaintext[32:]
    for i in range(16):
        L, R = des_round(L, R, keys[i])
        if verifyRound == i+1:
            verifyConcat = R + L

    encrypted_block = final_permutation(R + L)
    encrypted_block = binary_to_string(encrypted_block)

    if verifyRound == 0:
        return encrypted_block
    else:
        return encrypted_block, verifyConcat
```

This function takes the input block and the key, converts them into their respective binary values, and spawns 16 subkeys of length 48 bits from the key that has been supplied. Then it permutes the plaintext using the initial permutation table and splits it into two 32 bit halves. It then applies 16 rounds of the des algorithm and then finally swaps the two halves and applies the final permutation on it. Then it converts the resultant sequence of bits to a string and returns it.

It also has the capability to store and return the result of any particular round of the DES algorithm by specifying the parameter *verifyRound.*

3.1. DES Rounds -

The des_round() function applies the Mangler function on the right half by expanding it to 48 bits and then XORing with the round subkey. Then applying the S-boxes to shrink the output to 32 bits finally. It then XORs with the left half and returns the original right half with the new right half in the same order.

4. DES Decryption -
This function works in a similar manner like the encryption function. The only difference here is that the subkeys for all 16 rounds would be provided in the opposite manner, i.e., the 16th subkey woud be provided in the 1st round, 15th subkey would be provided in the 2nd round, etc.

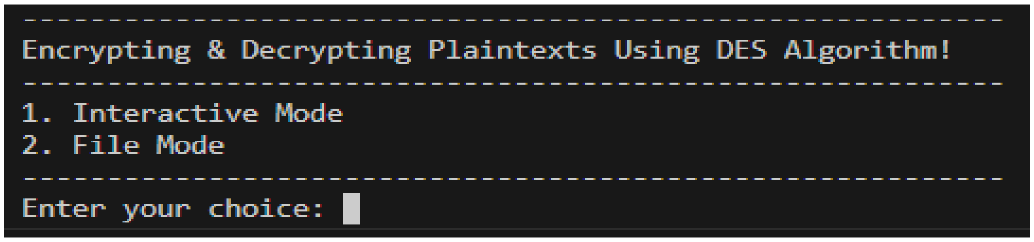
5. Verification -

This is done by a function called verify_rounds() which takes plaintext, ciphertext, encryption round, and the decryption round as an input parameters and uses the *verifyRound* parameter in the encryption and decryption functions to get the particular round values in binary. Then it uses assert statements to verify whether the results of round1 and round2 same or not. In case they are not same then it would raise an assertion error.

6. Program -

This program has two modes - Interactive and File Mode. Interactive mode asks you for a specific plaintext,ciphertext, key (or whatever is required) before executing that function while the File Mode takes the input of two files, i.e., plaintext.txt and keys.txt and then computes the encryption algorithm, decryption algorithm, and the verification for different rounds for each pair of <plaintext, key> or <ciphertext, key>.

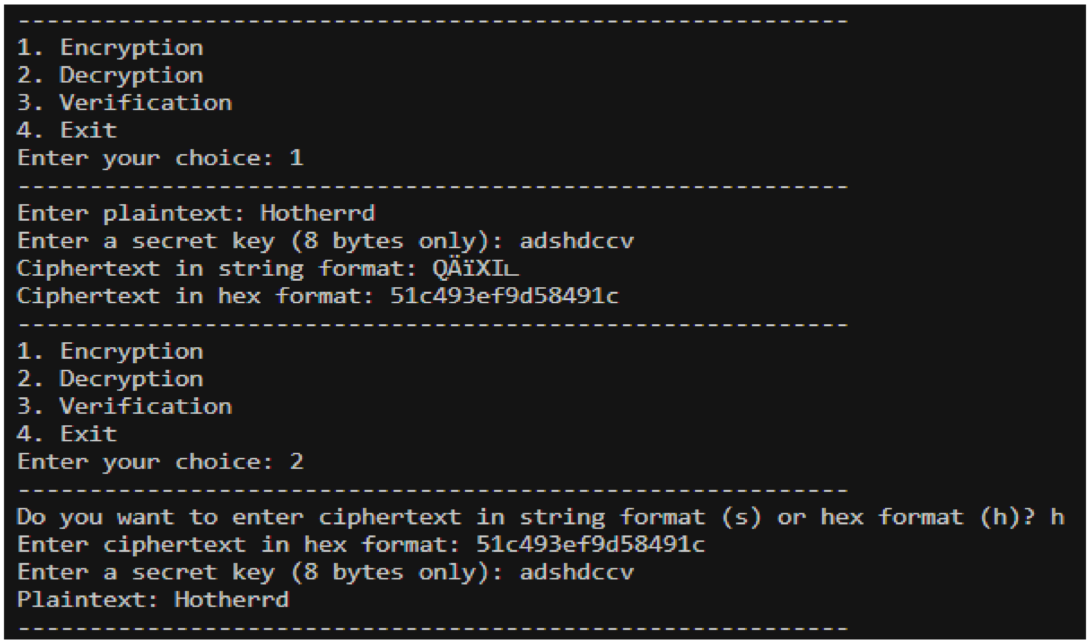
Main Menu -



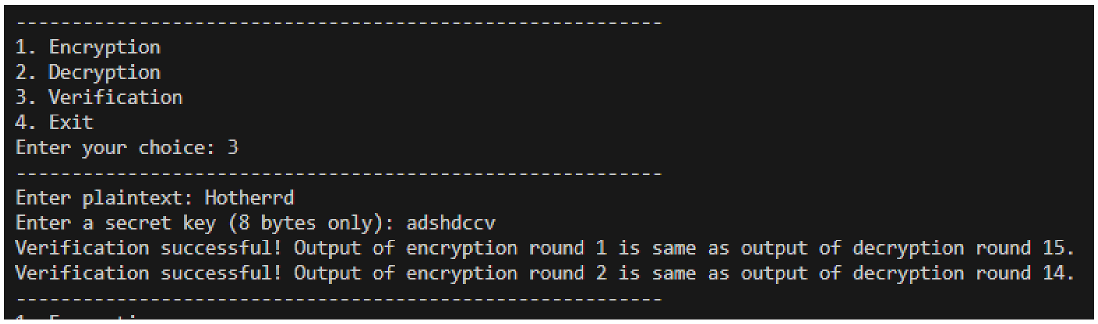
```
---------------------------------------------------------
Encrypting & Decrypting Plaintexts Using DES Algorithm!
---------------------------------------------------------
1. Interactive Mode
2. File Mode
---------------------------------------------------------
Enter your choice:
```

Mohammad Sufyan Azam, 2020312
Aamleen Ahmed, 2020002

Interactive Mode -

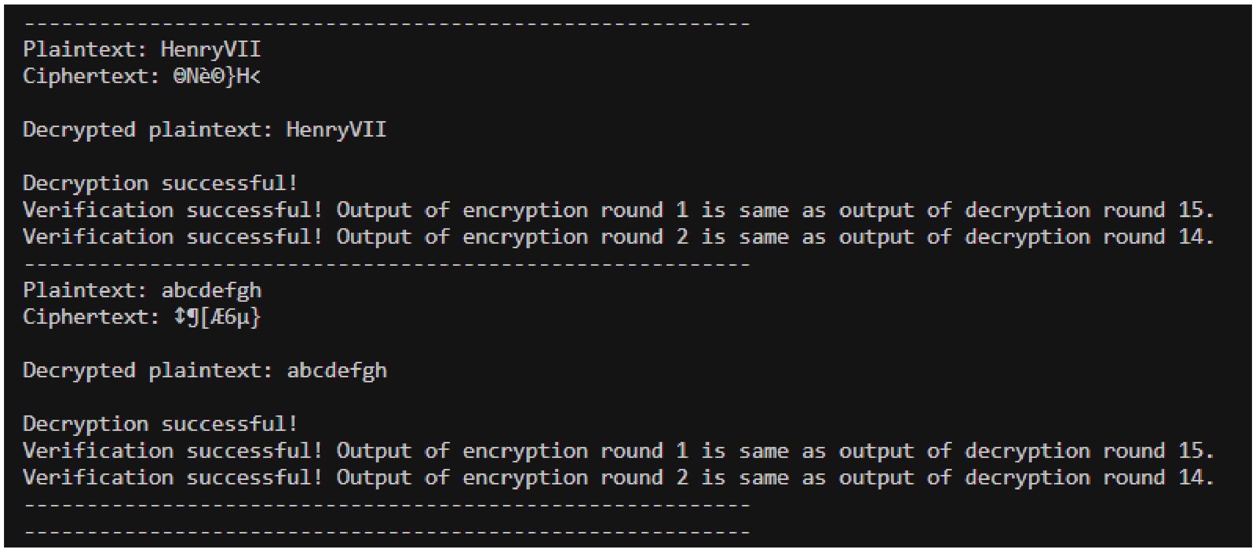


```
---------------------------------------------------------
1. Encryption
2. Decryption
3. Verification
4. Exit
Enter your choice: 1
---------------------------------------------------------
Enter plaintext: Hotherrd
Enter a secret key (8 bytes only): adshdccv
Ciphertext in string format: QÄïXI∟
Ciphertext in hex format: 51c493ef9d58491c
---------------------------------------------------------
1. Encryption
2. Decryption
3. Verification
4. Exit
Enter your choice: 2
---------------------------------------------------------
Do you want to enter ciphertext in string format (s) or hex format (h)? h
Enter ciphertext in hex format: 51c493ef9d58491c
Enter a secret key (8 bytes only): adshdccv
Plaintext: Hotherrd
---------------------------------------------------------
```



```
---------------------------------------------------------
1. Encryption
2. Decryption
3. Verification
4. Exit
Enter your choice: 3
---------------------------------------------------------
Enter plaintext: Hotherrd
Enter a secret key (8 bytes only): adshdccv
Verification successful! Output of encryption round 1 is same as output of decryption round 15.
Verification successful! Output of encryption round 2 is same as output of decryption round 14.
---------------------------------------------------------
```

File Mode -



```
---------------------------------------------------------
Plaintext: HenryVII
Ciphertext: ΘNèΘ}H<

Decrypted plaintext: HenryVII

Decryption successful!
Verification successful! Output of encryption round 1 is same as output of decryption round 15.
Verification successful! Output of encryption round 2 is same as output of decryption round 14.
---------------------------------------------------------
Plaintext: abcdefgh
Ciphertext: ↕¶[Æ6µ}

Decrypted plaintext: abcdefgh

Decryption successful!
Verification successful! Output of encryption round 1 is same as output of decryption round 15.
Verification successful! Output of encryption round 2 is same as output of decryption round 14.
---------------------------------------------------------
---------------------------------------------------------
```