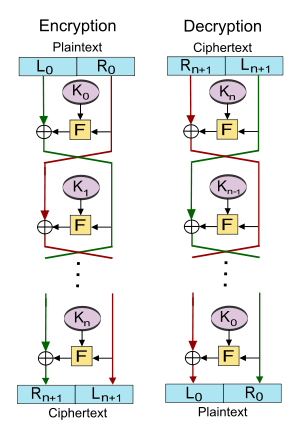
**AIM: Perform encryption and decryption using the following block cipher techniques (Feistel Cipher)**

**Feistel Cipher** model is a structure or a design used to develop many block ciphers such as DES. Feistel cipher may have invertible, non-invertible and self invertible components in its design. Same encryption as well as decryption algorithm is used. A separate key is used for each round. However same round keys are used for encryption as well as decryption.

**Feistel cipher algorithm**

1. Read list of all the Plain Text characters from the file (consider 2 character: block size)
2. Convert the Plain Text to Ascii and then 8-bit binary format.
3. Divide the binary Plain Text string into two halves: left half (L1) and right half (R1)
4. Generate a random binary keys (K1 and K2) of length equal to the half the length of the Plain Text for the two rounds.



**First Round of Encryption**

a. Generate function f1 using R1 and K1 as follows:

f1= xor(R1, K1) consider key k1: ‘A’

b. Now the new left half(L2) and right half(R2) after round 1 are as follows:

R2= xor(f1, L1)

L2=R1

**Second Round of Encryption**

a. Generate function f2 using R2 and K2 as follows:

f2= xor(R2, K2) consider key k1: ‘B’

b. Now the new left half(L3) and right half(R3) after round 2 are as follows:

R3= xor(f2, L2)

L3=R2

**Concatenation of R3 to L3 is the Cipher Text**

Same algorithm is used for decryption to retrieve the Plain Text from the Cipher Text.

**Note: Read the input from the file create a block of 16 bit i.e two characters at a time.**

**METHODOLOGY FOLLOWED:**

#include <iostream>

#include<bits/stdc++.h>

#include<fstream>

using namespace std;

string encryption(char L1,char R1,char K1,char K2){

    // first round

    char F1 = R1^K1;

    char R2 = F1^L1;

    char L2 = R1;

    //second round

    char F2 = R2^K2;

    char R3 =F2^L2;

    char L3 = R2;

    string st;

    st.push\_back(R3);

    st.push\_back(L3);

    return st;

}

string decryption(char L1,char R1,char K1,char K2){

    // first round

    char F1 = R1^K2;

    char R2 = F1^L1;

    char L2 = R1;

    //second round

    char F2 =R2^K1;

    char R3 =F2^L2;

    char L3 = R2;

    string st;

    st.push\_back(R3);

    st.push\_back(L3);

    return st;

}

int main()

{

    string st;

    char L1=' ';

    char R1=' ';

    char K1 = rand()%26+'A';

    char K2 = rand()%26+'A';

    cout<<"KEY1 : "<<K1<<"\n";

    cout<<"KEY2 : "<<K2<<"\n";

     ifstream fin;

     fin.open("input.txt");

    ofstream fout;

    fout.open("output.txt");

// encryption

 while(fin.get(L1) && fin.get(R1)){

        fout<<encryption(L1,R1,K1,K2);

  }

R1=' ';

fout<<encryption(L1,R1,K1,K2);

fin.close();

fout.close();

  ifstream fin2;

    fin2.open("output.txt");

    ofstream fout2;

    fout2.open("doutput.txt");

//decryption

while(fin2.get(L1) && fin2.get(R1)){

    fout2<<decryption(L1,R1,K1,K2);

}

fin.close();

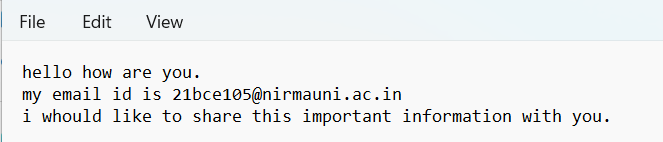
fout2.close();

    return 0;

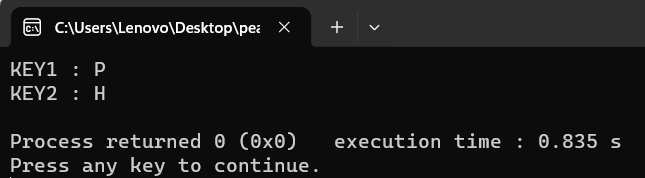
}

* **INPUT:**
* Here program gets Input from input.txt file

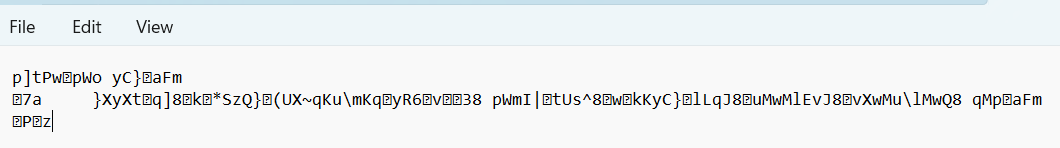
Key1 and key2 – randomly generate.



* Generated keys:



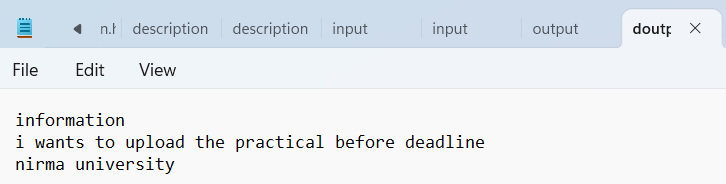
* After execution of the program, Encrypted message write in output.txt file



* For decryption,

Input from (encrypted message) - output.txt file

Output (decrypted message) - doutput.txt file



Note: this program done both the task -> (1) encryption and (2) decryption.