Fixed length collision resistant hash function code explanation:

- Here we generate two messages x1,x2 of length ‘N=16’ ,means input length is 32 bits and the hash function compresses it to 16 bits.
- We have some global variables GENERATOR,MODULUS,H_VAL which are used in DLP.
- MODULUS is a 16 bit prime number and GENERATOR ,H_VAL are primitive roots for that number.
- Let's call GENERATOR as G,H_VAL as H and MODULUS as q
- Now , first we will calculate G^x1 mod q and H^x2 mod q and we will multiply both and again will apply mod q.
- Padding will be done if it is necessary.

```
N = 16  # this is one of input length of hash function
GENERATOR = 6 #primitive roots for 60271
H_VAL = 7 #primitive roots for 60271
MODULUS = 60271 #this is 16 bit prime number
```

```
def FLHF(x1, x2):
    num1 = pow(GENERATOR, int(x1, 2), MODULUS)
    num2 = pow(H_VAL, int(x2, 2), MODULUS)
    ans = (num1*num2) % MODULUS
    ans = bin(ans).replace('0b', '').zfill(N)
    return ans
```