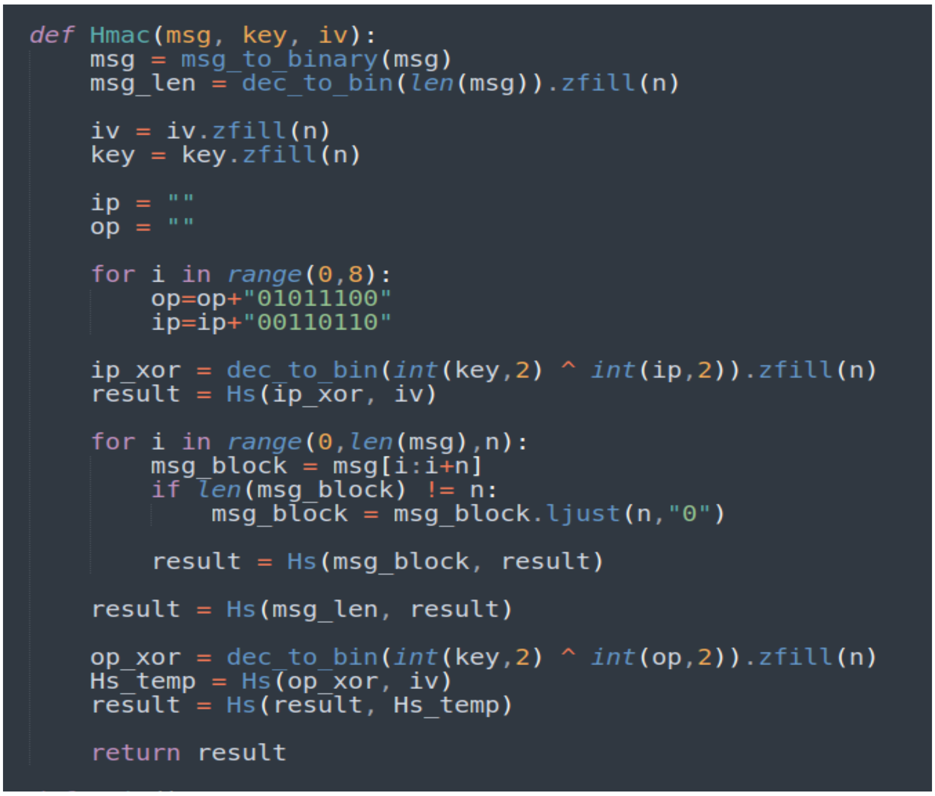
POIS ASSIGNMENT 1

TASK 8

USE COLLISION RESISTANT HASH FUNCTION TO BUILD H-MACS

THEORY



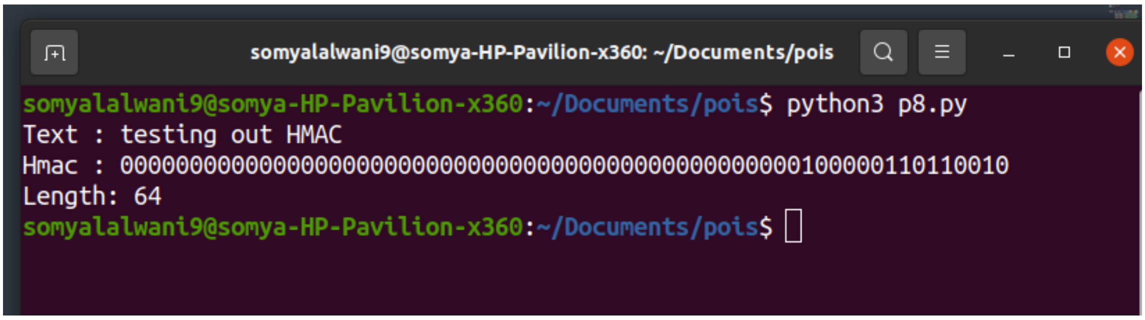
```python
def Hmac(msg, key, iv):
    msg = msg_to_binary(msg)
    msg_len = dec_to_bin(len(msg)).zfill(n)

    iv = iv.zfill(n)
    key = key.zfill(n)

    ip = ""
    op = ""

    for i in range(0,8):
        op=op+"01011100"
        ip=ip+"00110110"

    ip_xor = dec_to_bin(int(key,2) ^ int(ip,2)).zfill(n)
    result = Hs(ip_xor, iv)

    for i in range(0,len(msg),n):
        msg_block = msg[i:i+n]
        if len(msg_block) != n:
            msg_block = msg_block.ljust(n,"0")

        result = Hs(msg_block, result)

    result = Hs(msg_len, result)

    op_xor = dec_to_bin(int(key,2) ^ int(op,2)).zfill(n)
    Hs_temp = Hs(op_xor, iv)
    result = Hs(result, Hs_temp)

    return result
```

Here, Hs stands for Hashing function,
msg is the original message
ip,op is ipad,opad from the definition
ip_xor and op_xor are input and output signatures respectively,
msg_block is the ith block in original message msg, where i ranges from [1, len(msg))
key is the secret key used for hashing
iv is an initial vector (some constant)

Pseudo Code :

```
ip_xor = padded_key ⊕ip
op_xor = padded_key ⊕op
Initialising result :
        result = Hs(ip_xor, iv) (Hash function)
For each msg block :
        result = Hs(msg_block, result)
Hs_temp = Hs(op_xor, iv)
result = Hs(result, Hs_temp)
```

OUTPUT :



```
somyalalwani9@somya-HP-Pavilion-x360:~/Documents/pois$ python3 p8.py
Text : testing out HMAC
Hmac : 0000000000000000000000000000000000000000000000000100000110110010
Length: 64
somyalalwani9@somya-HP-Pavilion-x360:~/Documents/pois$ 
```