POIS ASSIGNMENT 1

TASK 7

USE MERKLE-DAMGARD TRANSFORM TO OBTAIN A PROVABLY SECURE COLLISION RESISTANT HASH FUNCTION

CODE

```
n =64
def merkle_hash(msg, iv):
    msg = msg_to_binary(msg)
    l = len(msg)
    msg_len = dec_to_bin(l).zfill(n)
    result = iv.zfill(n)
    for i in range(0,l,n):
        cur_msg=msg[i:i+n]
        if n != len(cur_msg):
            cur_msg = cur_msg.ljust(n,"0")
        result=Hs(cur_msg, result)

    result = Hs(msg_len,result)
    return result
```

As stated in the theory, the “msg” is split into blocks of constant length=n=64.
Chunks are if necessary, padded.
Set result = an initial value of iv
And the result is calc again and again using the hash function result of previous chunk.

result=Hs(cur_msg, result)

(h_{i+1} = H~(h_i; x_i) for i > 1. Then, H(x) = h_n+1 (b bits))

Hs is the hash function of previous code.

OUTPUT :

```
somyalalwani9@somya-HP-Pavilion-x360: ~/Documents/pois
somyalalwani9@somya-HP-Pavilion-x360:~/Documents/pois$ python3 p7.py
Text: test string for merkle hash
enc string:  0000000000000000000000000000000000000000000000000000000101110001111010
Length:64
somyalalwani9@somya-HP-Pavilion-x360:~/Documents/pois$
```