Filename: hmac.py
Running instruction: `python3 hmac.py <plaintext> <seed>`
Input: plaintext, seed
Class: HMAC
Function: generate

The DLP_HASH class is imported and used here directly. The ipad and opad will be initialised according to the standard values. The generate function will first pad these ipad and opad values by repeating their values as many times as required and then calculate the XOR of seed and the ipad. The XORed value and an initialisation vector will be given as input to the DLP_HASH object. Inside a loop, a block of plaintext will be extracted and will be given as input to the DLP_HASH object along with the previous output. The output will be obtained and again this output and another block of plaintext will be given to the DLP_HASH to obtain an output. After the loop terminates, the length of the message(encoded as bit string) and the output will again be given to the DLP_HASH. Let's call the output now generated as output1. Another output, called output2 will be obtained by giving the output1 and the initialisation vector to the DLP_HASH object. Finally these output1 and output2 will be given as input again to another DLP_HASH object to obtain the final output.