SSN College of Engineering,
Department of Computer Science and Engineering
IT 8761 Security Laboratory

Exercise 8:

To implement the message digest SHA1

Programming Language: Java

Hints:

1. Read the message
2. Divide the message into 512 bit blocks.
3. Append padding bits,
    - A single "1" bit is appended to the message, and then "0" bits are appended so that the length in bits of the padded message equals to 448 mod 512.
4. Append length
    - A 64-bit representation of the length of the message is appended
5. Initialize MD buffers, A, B, C, D,E
    - word A: 67452301
    - word B: efcdab89
    - word C: 98badcfe
    - word D: 10325476
    - word E: c3d2e1f0
6. Invoke the compress function for four times
7. Display the message digest from the buffers.

SHA1 Compression function:

1. (A,B,C,D,E) <- (E+f(t,B,C,D)+(A<< 5)+W_t+K t), A, (B<<30), C, D)

   - t is the step number
   - W_t is derived from the message block
   - K_t is a constant value derived from sin table.
     - *K(t) = 5A827999 (0 <= t <= 19)*
     - *K(t) = 6ED9EBA1 (20 <= t <= 39)*
     - *K(t) = 8F1BBCDC (40 <= t <= 59)*
     - *K(t) = CA62C1D6 (60 <= t <= 79)*
   - Each f(t), 0 <= t <= 79, operates on three 32-bit words B, C, D and produces a 32-bit word as output.
     - f(t;B,C,D) is defined as follows: for words B, C, D,
     - f(t;B,C,D) = (B *AND* C) OR ((*NO*T B) *AND* D) (0 <= t <= 19)
     - f(t;B,C,D) = B *XOR* C *XOR* D (20 <= t <= 39)
     - f(t;B,C,D) = (B *AND* C) *OR* (B *AND* D) *OR* (C *AND* D) (40 <= t <= 59)
     - f(t;B,C,D) = B *XOR* C *XOR* D (60 <= t <= 79).
   - <<<s circular left shift of 32 bit argument by s bits
   - + is addition modulo 2^{32}

2. Perform a 32 bit circular right shift such that;

   - a=e; b=a; c=b; d=c; e=d;

<u>For further clarification, read the associated pdf</u>

T[1] = D76AA478	T[17] = F61E2562	T[33] = FFFA3942	T[49] = F4292244
T[2] = E8C7B756	T[18] = C040B340	T[34] = 8771F681	T[50] = 432AFF97
T[3] = 242070DB	T[19] = 265E5A51	T[35] = 699D6122	T[51] = AB9423A7
T[4] = C1BDCEEE	T[20] = E9B6C7AA	T[36] = FDE5380C	T[52] = FC93A039
T[5] = F57COFAF	T[21] = D62F105D	T[37] = A4BEEA44	T[53] = 655B59C3
T[6] = 4787C62A	T[22] = 02441453	T[38] = 4BDECFA9	T[54] = 8F0CCC92
T[7] = A8304613	T[23] = D8A1E681	T[39] = F6BB4B60	T[55] = FFEFF47D
T[8] = FD469501	T[24] = E7D3FBC8	T[40] = BEBFBC70	T[56] = 85845DD1
T[9] = 698098D8	T[25] = 21E1CDE6	T[41] = 289B7EC6	T[57] = 6FA87E4F
T[10] = 8B44F7AF	T[26] = C33707D6	T[42] = EAA127FA	T[58] = FE2CE6E0
T[11] = FFFF5BB1	T[27] = F4D50D87	T[43] = D4EF3085	T[59] = A3014314
T[12] = 895CD7BE	T[28] = 455A14ED	T[44] = 04881D05	T[60] = 4E0811A1
T[13] = 6B901122	T[29] = A9E3E905	T[45] = D9D4D039	T[61] = F7537E82
T[14] = FD987193	T[30] = FCEFA3F8	T[46] = E6DB99E5	T[62] = BD3AF235
T[15] = A679438E	T[31] = 676F02D9	T[47] = 1FA27CF8	T[63] = 2AD7D2BB
T[16] = 49B40821	T[32] = 8D2A4C8A	T[48] = C4AC5665	T[64] = EB86D391

T table