Minimal SHA-1 Implementation Report

1 Introduction

This report describes a minimal C++ implementation of the SHA-1 cryptographic hash function. The code computes the SHA-1 digest of an input message and prints the final 160-bit (20-byte) hash in hexadecimal form

2 Overview of SHA-1

SHA-1 (Secure Hash Algorithm 1) is a 160-bit hashing algorithm standardized by NIST. It processes the input message in 512-bit blocks using a series of logical functions, word expansions, and modular additions. The output digest consists of 5 words:

$$H0,H1,H2,H3,H4$$

which together form a 40-digit hexadecimal string.

3 Message Padding

SHA-1 uses a specific padding scheme:

1. Append a single "1" bit (0x80).
2. Append zero bits until the message length is congruent to 56 (mod 64).
3. Append the original message length as a 64-bit big-endian integer.

The implementation follows these rules in the sha1.pad() function

4 Message Schedule

Each 512-bit block is divided into 16 initial 32-bit words:

$$W0,W1,\ldots,W15$$

The remaining 64 words are generated as:

$$Wt = ROTL1(Wt-3 \oplus Wt-8 \oplus Wt-14 \oplus Wt-16) \text{ for } t = 16\ldots79$$

5 Compression Function

The algorithm maintains five working variables:

A=H0, B =H1, C =H2, D=H3, E =H4

For each round t, one of the four SHA-1 nonlinear functions is chosen depending on the round index:

$$f(t) = \begin{cases} (B \wedge C) \vee (\neg B \wedge D), & t < 20 \\ B \oplus C \oplus D, & t < 40 \\ (B \wedge C) \vee (B \wedge D) \vee (C \wedge D), & t < 60 \\ B \oplus C \oplus D, & t < 80 \end{cases}$$

The working variables are updated as:

TEMP=ROTL5(A)+f(t)+E+Wt+Kt

E =D, D=C, C=ROTL30(B), B =A, A=TEMP

After processing all 80 rounds:

H0 =H0+A, H1 =H1+B, H2 =H2+C, H3 =H3+D, H4=H4+E

6 Final Digest

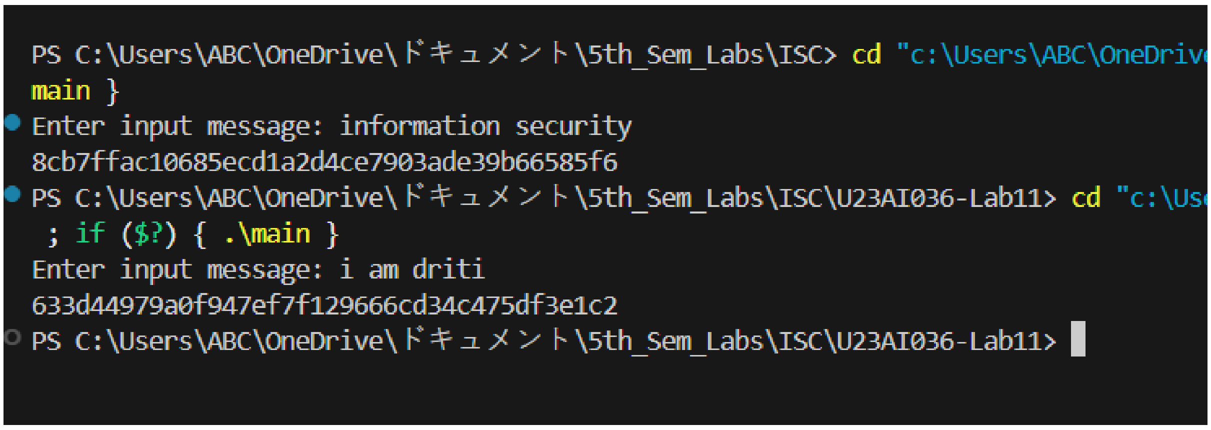
The final 160-bit digest is produced by concatenating:

H0 ∥H1 ∥H2 ∥H3 ∥H4

Each 32-bit word is output in big-endian byte order to produce the standard 40-hex-digit SHA-1 hash

7 Output

The program converts the 20-byte digest into a hexadecimal string via the bytes function, and prints it to standard output



```
PS C:\Users\ABC\OneDrive\ドキュメント\5th_Sem_Labs\ISC> cd "c:\Users\ABC\OneDriv
main }
Enter input message: information security
8cb7ffac10685ecd1a2d4ce7903ade39b66585f6
PS C:\Users\ABC\OneDrive\ドキュメント\5th_Sem_Labs\ISC\U23AI036-Lab11> cd "c:\Us
 ; if ($?) { .\main }
Enter input message: i am driti
633d44979a0f947ef7f129666cd34c475df3e1c2
PS C:\Users\ABC\OneDrive\ドキュメント\5th_Sem_Labs\ISC\U23AI036-Lab11>
```

8 Conclusion

This implementation provides a compact, educational version of the SHA-1 hashing algorithm, cov ering padding, message expansion, the 80-round compression function, and final digest formatting