Practical Number 1

Aim: To implement Affine Cipher.

Theory:

Affine cipher is a monoalphabetic cipher in which relationship between a symbol in the plaintext to a symbol in the ciphertext is always one-to-one i.e. every occurrence of the plaintext character is replaced by the same character in the cipher text. Affine cipher uses combination of additive cipher and multiplicative cipher with a pair of key. The encryption and decryption process can be expressed as:

$$C = ((P * K_1) + K_2) \bmod 26$$

$$P = ((C – K_2) * K_1^{-1}) \bmod 26$$

Where K_1^{-1} is the multiplicative inverse of K_1 and $–K_2$ is the additive inverse of K_2

The Additive cipher is a special case of an Affine cipher in which $k_1 = 1$. The Multiplicative cipher is a special case of an Affine cipher in which $k_2 = 0$.

Monoalphabetic ciphers are vulnerable to ciphertext-only attacks using Brute –force attacks. Monoalphabetic ciphers are also subject to statistical attacks (e.g. language character frequency). Another way to attack is to know the occurrence of specific letter combinations, i.e. know the frequency of two-letter (diagrams) or three-letter (trigrams) strings in the ciphertext and compare them with the frequency of two-letter or three-letter strings in the underlying language of the plaintext.
Apart from brute-force and statistical method of ciphertext-only attack, an affine cipher can also be attacked with chosen-plaintext attack.

Example: Encrypt the message “hello” with the key pair (7, 2)
The multiplicative key is 7 and the additive key is 2

Plaintext	Encryption	Ciphertext
h → 07	(07 * 07 + 2) mod 26	25 → Z
e → 04	(04 * 07 + 2) mod 26	04 → E
l → 11	(11 * 07 + 2) mod 26	01 → B
l → 11	(11 * 07 + 2) mod 26	05 → B
o → 14	(14 * 07 + 2) mod 26	22 → W

hello → ZEBBW

Ciphertext	Decryption	Plaintext
Z → 25	((25 - 2) * 7_{-1})mod 26	07 → h
E→ 04	((04 - 2) * 7_{-1})mod 26	04 → e
B → 01	((01 - 2) * 7_{-1})mod 26	11 → l
B → 01	((01 - 2) * 7_{-1})mod 26	11 → l

W→ 22	((22 - 2) * 7_{-1})mod 26	14 → o

ZEBBW→ hello

Conclusion: Affine cipher has been studied and implemented.