5.Plain text

Program:

#include <stdio.h>

int mod\_inverse(int a, int m) {

for (int x = 1; x < m; x++) {

if ((a \* x) % m == 1) {

return x;

}

}

return -1; // Modular inverse doesn't exist

}

int affine\_cipher(int a, int b, int p) {

return (a \* p + b) % 26;

}

int main() {

int a, b;

char plaintext[100];

char ciphertext[100];

printf("Enter the value of a: ");

scanf("%d", &a);

printf("Enter the value of b: ");

scanf("%d", &b);

if (mod\_inverse(a, 26) == -1) {

printf("Modular inverse of 'a' doesn't exist. Choose a different 'a' value.\n");

return 1;

}

printf("Enter the plaintext: ");

scanf(" %[^\n]s", plaintext);

int i = 0;

while (plaintext[i]) {

if (plaintext[i] >= 'A' && plaintext[i] <= 'Z') {

ciphertext[i] = 'A' + affine\_cipher(a, b, plaintext[i] - 'A');

} else if (plaintext[i] >= 'a' && plaintext[i] <= 'z') {

ciphertext[i] = 'a' + affine\_cipher(a, b, plaintext[i] - 'a');

} else {

ciphertext[i] = plaintext[i];

}

i++;

}

ciphertext[i] = '\0';

printf("Ciphertext: %s\n", ciphertext);

return 0;

}

Output:

