***SAVEETHA SCHOOL OF ENGINEERING***

***SAVEETHA INSTITUTE OF MEDICAL AND TECHNICAL SCIENCE***

**EXP NO 16: Write a C program for ciphertext has been generated with an affine cipher. The most frequent letter of the ciphertext is “B,” and the second most frequent letter of the ciphertext is “U.”Break this code.**

**AIM**

To Write a C program for ciphertext has been generated with an affine cipher. The most frequent letter of the ciphertext is “B,” and the second most frequent letter of the ciphertext is “U.”Break this code.

**PROCEDURE**

* Download and install any c application
* Traverse the given text one character at a time .
* For each character, transform the given character as per the rule,depending on whether we’re encrypting or decrypting the text.
* Return the new string generated.

**PROGRAM**

#include <stdio.h>

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

int m0 = m, t, q;

int x0 = 0, x1 = 1;

if (m == 1)

return 0;

while (a > 1) {

q = a / m;

t = m;

m = a % m, a = t;

t = x0;

x0 = x1 - q \* x0;

x1 = t;

}

if (x1 < 0)

x1 += m0;

return x1;

}

int main() {

char most\_frequent = 'B';

char second\_most\_frequent = 'U';

int most\_frequent\_index = most\_frequent - 'A';

int second\_most\_frequent\_index = second\_most\_frequent - 'A';

// Assuming English alphabet, which has 26 letters

int m = 26;

// Calculate the difference between the letters in the ciphertext

int diff\_most\_second = (most\_frequent\_index - second\_most\_frequent\_index + m) % m;

// Find the multiplicative inverse of the difference

int inverse\_diff = mod\_inverse(diff\_most\_second, m);

// Assuming the additive key is 0 (it's not provided in the problem statement)

int additive\_key = 0;

// Decipher the ciphertext

printf("Deciphered Text:\n");

// Assuming the ciphertext is stored in the variable 'ciphertext'

char ciphertext[] = "CIPHERTEXT\_TO\_DECRYPT";

for (int i = 0; ciphertext[i] != '\0'; i++) {

char encrypted\_char = ciphertext[i];

if (encrypted\_char >= 'A' && encrypted\_char <= 'Z') {

int encrypted\_index = encrypted\_char - 'A';

int decrypted\_index = ((inverse\_diff \* (encrypted\_index - additive\_key)) % m + m) % m;

char decrypted\_char = 'A' + decrypted\_index;

printf("%c", decrypted\_char);

} else {

printf("%c", encrypted\_char); // Print non-alphabetic characters as is

}

}

printf("\n");

return 0;

}

**OUTPUT**

