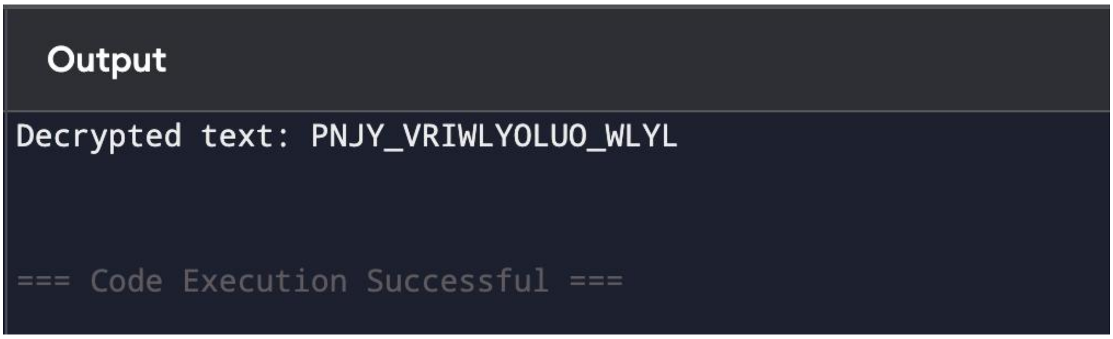
EXP-6:-

PROGRAM:

```
#include <stdio.h>
#include <string.h>

#define ALPHABET_SIZE 26

// Function to find the modular inverse of a
int modInverse(int a, int m) {
    for (int x = 1; x < m; x++) {
        if ((a * x) % m == 1) {
            return x;
        }
    }
    return -1;
}

// Function to decrypt the ciphertext
void decrypt(char *ciphertext, int a, int b) {
    int m = ALPHABET_SIZE;
    int a_inv = modInverse(a, m);
    if (a_inv == -1) {
        printf("No modular inverse found for a = %d\n", a);
        return;
    }

    for (int i = 0; ciphertext[i] != '\0'; i++) {
        if (ciphertext[i] >= 'A' && ciphertext[i] <= 'Z') {
            ciphertext[i] = (a_inv * (ciphertext[i] - 'A' - b + m)) % m + 'A';
        }
    }
}

int main() {
    char ciphertext[] = "YOUR_CIPHERTEXT_HERE"; // Replace with actual ciphertext
    int a = 5; // Example value for 'a'
    int b = 1; // Example value for 'b'

    decrypt(ciphertext, a, b);
    printf("Decrypted text: %s\n", ciphertext);
    return 0;
}
```

OUTPUT:



```
Decrypted text: PNJY_VRIWLYOLUO_WLYL



=== Code Execution Successful ===
```