***SAVEETHA SCHOOL OF ENGINEERING***

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

**EXP NO 7 Write a program for demonstration of encrypting and decrypting the messages using the Blow Fish algorithm logic.**

**AIM**

To Write a program for demonstration of encrypting and decrypting the messages using the Blow Fish algorithm logic.

**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>

// Function to calculate modular exponentiation

unsigned long long mod\_exp(unsigned long long base, unsigned long long exp, unsigned long long mod) {

unsigned long long result = 1;

base = base % mod;

while (exp > 0) {

if (exp % 2 == 1)

result = (result \* base) % mod;

exp = exp >> 1;

base = (base \* base) % mod;

}

return result;

}

// Function to encrypt plaintext using RSA public key

unsigned long long rsa\_encrypt(unsigned long long plaintext, unsigned long long e, unsigned long long n) {

return mod\_exp(plaintext, e, n);

}

// Function to decrypt ciphertext using RSA private key

unsigned long long rsa\_decrypt(unsigned long long ciphertext, unsigned long long d, unsigned long long n) {

return mod\_exp(ciphertext, d, n);

}

int main() {

unsigned long long p, q, n, phi, e, d;

unsigned long long plaintext, encrypted, decrypted;

// Get user input for RSA parameters

printf("Enter value for p (prime number): ");

scanf("%llu", &p);

printf("Enter value for q (prime number): ");

scanf("%llu", &q);

printf("Enter value for e (public exponent): ");

scanf("%llu", &e);

// Calculate n, phi, and d

n = p \* q;

phi = (p - 1) \* (q - 1);

// Find the modular inverse of e mod phi

for (d = 1; d < phi; ++d) {

if ((e \* d) % phi == 1) {

break;

}

}

// Get user input for plaintext

printf("Enter plaintext to be encrypted: ");

scanf("%llu", &plaintext);

// Encrypt plaintext

encrypted = rsa\_encrypt(plaintext, e, n);

// Decrypt ciphertext

decrypted = rsa\_decrypt(encrypted, d, n);

// Output results

printf("Plaintext: %llu\n", plaintext);

printf("Encrypted: %llu\n", encrypted);

printf("Decrypted: %llu\n", decrypted);

return 0;

}

**OUTPUT**

