

Name: Saeed Alabdullah
ID:1935229

Assignment task 2

Code:

```
#include <stdio.h>
#include <string.h>
#include <openssl/bn.h>
#include <stdlib.h>
#include <unistd.h>
#include <sys/wait.h>

void printBN(char *msg, BIGNUM *tmp) {
    char *number_str = BN_bn2hex(tmp); // Convert BIGNUM to hex
    printf("%s%s\n", msg, number_str); // Print hex
    OPENSSL_free(number_str); // Free memory
}

int main(int argc, char *argv[]) {
    BN_CTX *ctx = BN_CTX_new();

    // Initialize all needed BIGNUM variables
    BIGNUM *e = BN_new();
    BIGNUM *d = BN_new();
    BIGNUM *n = BN_new();
    BIGNUM *phi_n = BN_new();
    BIGNUM *C = BN_new();
    BIGNUM *D = BN_new();

    // Assign values (replace placeholders with actual values)
```

Name: Saeed Alabdullah

ID:1935229

```
BN_hex2bn(&e, "010001"); // Placeholder: Replace with actual e value

BN_hex2bn(&n, "E103ABD94892E3E74AFD724BF28E78366D9676BCCC70118BD0AA1968DBB143D1");
// Placeholder: Replace with actual n value

BN_hex2bn(&phi_n,
"E103ABD94892E3E74AFD724BF28E78348D52298BD687C44DEB3A81065A7981A4"); // Placeholder:
Replace with actual phi_n value

BN_hex2bn(&C, "0123456789ABCDEF"); // Placeholder: Replace with actual Ciphertext value

// Calculate the Decryption Key (Private Key) d = e^-1 mod (phi_n)

BN_mod_inverse(d, e, phi_n, ctx);

// Decrypt Ciphertext using D = C^d mod n

BN_mod_exp(D, C, d, n, ctx);

// Print the Decryption Key

printBN("Decryption Key (d): ", d);

// Print the Decrypted Ciphertext

printBN("Decrypted Ciphertext (D): ", D);

// Convert Hex string to ASCII letters

printf("\nOriginal Message:\n");

char str1[500] = "print(\"";

char *str2 = BN_bn2hex(D);

char str3[] = "\".decode(\"hex\")\"";

strcat(str1, str2);

strcat(str1, str3);

// Run Python command to print the original message

char *args[] = {"python2", "-c", str1, NULL};

execvp("python2", args);

return EXIT_SUCCESS;

}
```

Name: Saeed Alabdullah

ID:1935229

To use the previous code , you will need to get your own public and private keys using the following code:

```
openssl genpkey -algorithm RSA -out private.pem
```

```
openssl rsa -pubout -in private.pem -out public.pem
```

first is to get the private key, and the 2nd is to separate your public key, then you can use cat command to view it as follows:

```
-----END PRIVATE KEY-----
saeed@lamp ~$ cat public.pem
-----BEGIN PUBLIC KEY-----
MIIBIjANBgkqhkiG9w0BAQEFAAOCAQ8AMIIBCgKCAQEAuyXMFML6uavo3WExdm+7
q/nUq56JjFw0Nu2ew02Byi7J79zW9m1shTUMoD3t10fKIkei2JXo6aw5w0CF3uZ
Ku00lumcv8eT1Q2e iq1PG2gtQKIWkLciPQQhjH2G9B4oEs7McUQm6X8vnJuAp8+3
1NoN2D/t8hUG2aZhh/bdD2ehnm51iWa/ba02rbUJrUXdNSXsqFD58EuyoTy/Lk/j
Qt0h3kzzUw4P/Uaq1xRcAu i5Y/5NpFAyC3C10fWCO2Q9SbMDULbBmbP2FJ4HEIdW
IT3zX5IJtxrbY5ARpr68+YH7Y4cSN1KTjA0ryiUEAV0I1Xf+bM5zKjcw7UjjJHBF
HQIDAQAB
-----END PUBLIC KEY-----
saeed@lamp ~$
```