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: