Name: Moayad Talal Alghamdi

ID: 2035408

Instructor: Turki Abdulhafiz

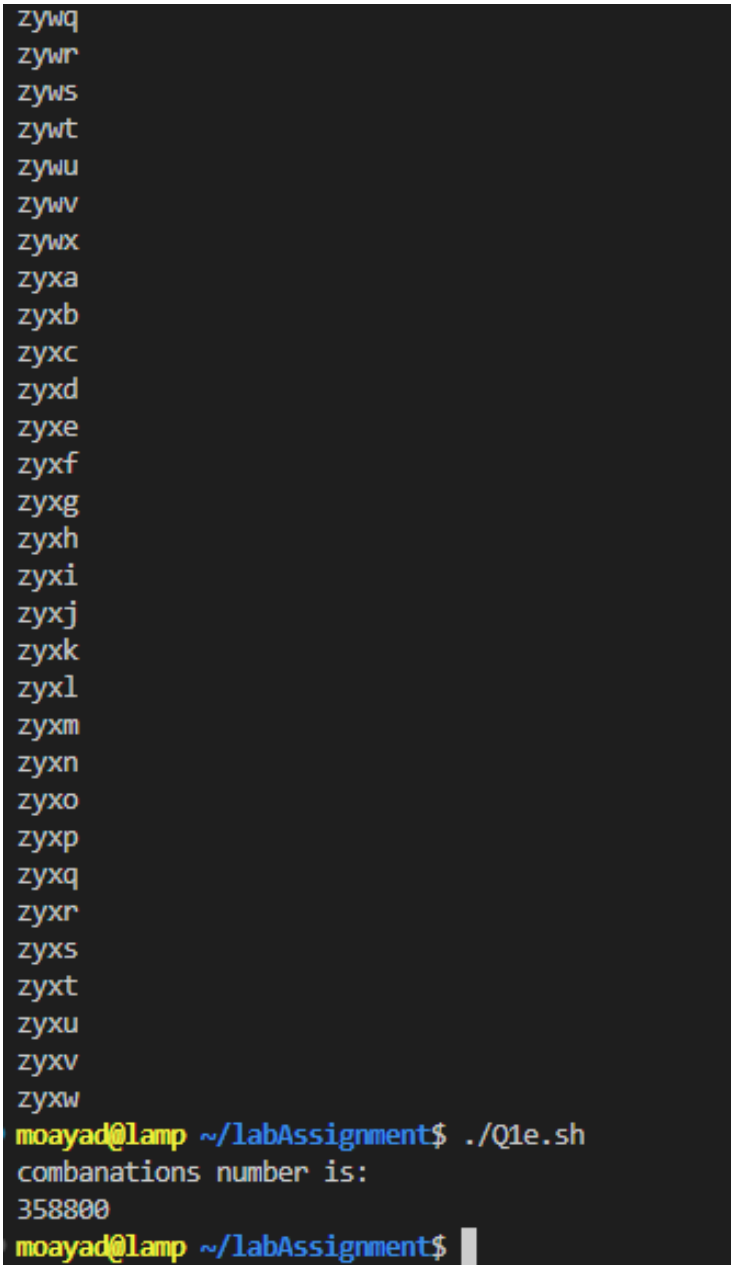
Q1 Source code)

```c
#include <stdio.h>

int main (int argc, char* argv[]){

    char character[] = "abcdefghijklmnopqrstuvwxyz";
    int ch1;
    int ch2;
    int ch3;
    int ch4;

    for(ch1 = 0; ch1 < 26; ch1++){
        for(ch2 = 0; ch2 < 26; ch2++){
                if(ch2 == ch1){
                    continue;
                }
            for(ch3 = 0; ch3 < 26; ch3++){
                if(ch3 == ch1 || ch3 == ch2 ){
                    continue;
                }
                for(ch4 = 0; ch4 < 26; ch4++){
                    if(ch4 == ch1 || ch4 == ch2 || ch4 == ch3 ){
                        continue;
                    }

                    printf("%c%c%c%c\n",character[ch1],character[ch2],character[ch3],character[ch4]);
                }
            }
        }
    }



    return 0;
}
```

Q1 shell code)

```
echo "combanations number is:"
./Q1 | wc -l
```

Q1 Output)

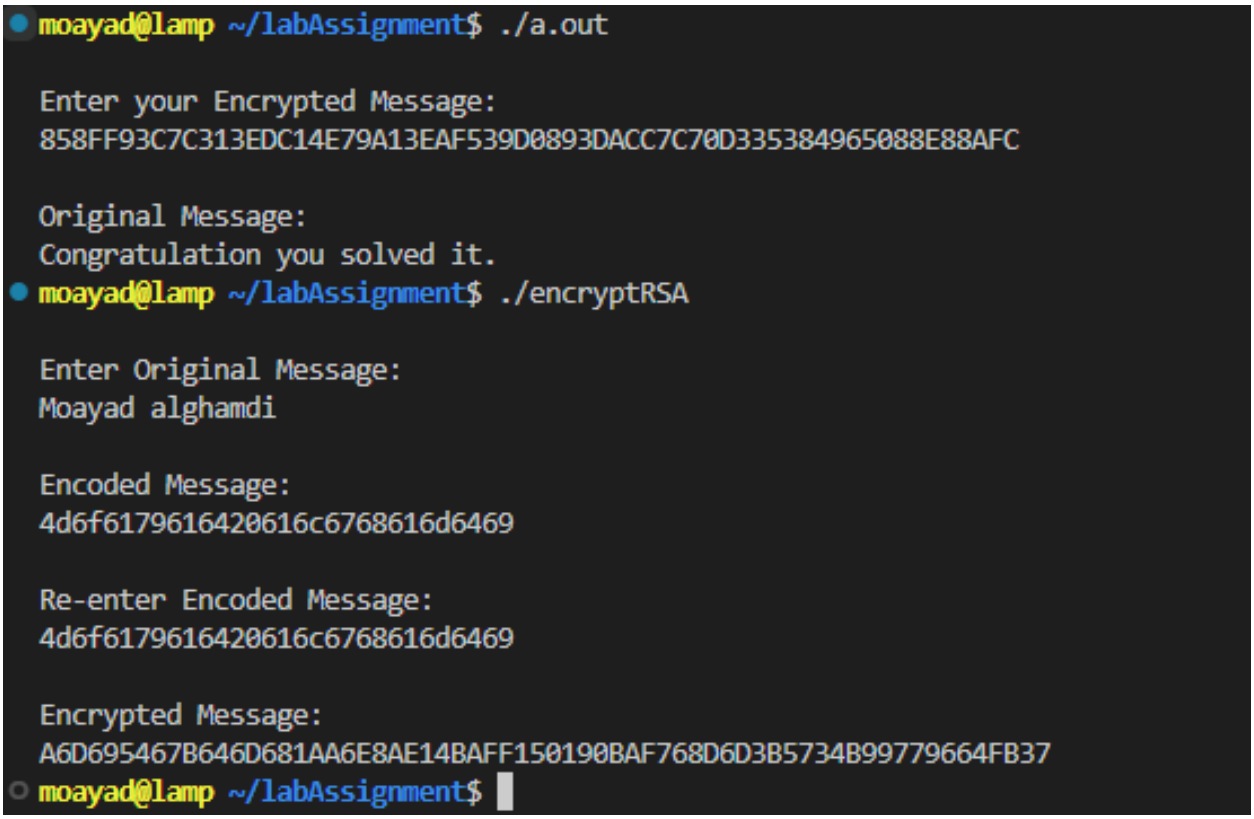

zywq
zywr
zyws
zywt
zywu
zywv
zywx
zyxa
zyxb
zyxc
zyxd
zyxe
zyxf
zyxg
zyxh
zyxi
zyxj
zyxk
zyxl
zyxm
zyxn
zyxo
zyxp
zyxq
zyxr
zyxs
zyxt
zyxu
zyxv
zyxw
moayad@lamp ~/labAssignment$ ./Q1e.sh
combanations number is:
358800
moayad@lamp ~/labAssignment$

Q2 Source code)

```c

#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();

////////////////////////////////////////////////////////////

// Here initialize all needed BIGNUM variables
// 1- Encryption Key variable
// 2- Decryption Key variable
// 3- product of large prime numbers p and q
// 4- Totient of (n) Euler's totient function
// 5- Encrypted Message variable
// 6- Decrypted Ciphertext variable
BIGNUM *encry = BN_new();
BIGNUM *dencry = BN_new();
BIGNUM *pPQ = BN_new();
BIGNUM *tot = BN_new();
BIGNUM *encryMessage = BN_new();
BIGNUM *decCiphertext = BN_new();

////////////////////////////////////////////////////////////

// Find Decryption Key (d) using (e) and (Phin):
// 1- Assign value to (e) Encryption Key from hex
// 2- Assign value to (Phin) Encryption Key from hex
// 3- Calculate the Decryption Key (Private Key) d=e mod(Phi(n))

BN_hex2bn(&encry, "010001");
BN_hex2bn(&tot, "E103ABD94892E3E74AFD724BF28E78348D52298BD687C44DEB3A81065A7981A4");
BN_mod_inverse(dencry, encry, tot, ctx);

```

```
44   char *CC= malloc(100 * sizeof(char));
45   printf("\nEnter your Encrypted Message:\n");
46   // Read the Encrypted Message from the user to variable CC
47   fgets(CC, 100, stdin);
48   // Assign the input value in variable (CC) to Encrypted Message variable
49   BN_hex2bn(&encryMessage, CC);
50
51
52   /*
53   Decrypt ciphertext using D=C^d(mod(n)) ,
54   where: (D) is the Decrypted Ciphertext and (C) is the Ciphertext
55   */
56   // Assign value to (n) product of two large prime numbers from hex
57   BN_hex2bn(&pPQ, "E103ABD94892E3E74AFD724BF28E78366D9676BCCC70118BD0AA1968DBB143D1");
58   // decrypt Ciphertext using the Private Key
59   BN_mod_exp(decCiphertext, encryMessage, dencry, pPQ, ctx);
60
61   // Convert Hex string to ASCII letters
62   printf("\nOriginal Message:\n");
63   char str1[500]="print(\"";
64   char *str2 = BN_bn2hex(decCiphertext);
65   char str3[]="\".decode(\"hex\"))";
66   strcat(str1,str2);
67   strcat(str1,str3);
68   char* args[]={"python2", "-c",str1, NULL};
69   execvp("python2", args);
70   return EXIT_SUCCESS;
71   }
```

Q2 output)


moayad@lamp ~/labAssignment$ ./a.out
Enter your Encrypted Message:
858FF93C7C313EDC14E79A13EAF539D0893DACC7C70D335384965088E88AFC
Original Message:
Congratulation you solved it.
moayad@lamp ~/labAssignment$ ./encryptRSA
Enter Original Message:
Moayad alghamdi
Encoded Message:
4d6f6179616420616c6768616d6469
Re-enter Encoded Message:
4d6f6179616420616c6768616d6469
Encrypted Message:
A6D695467B646D681AA6E8AE14BAFF150190BAF768D6D3B5734B99779664FB37
moayad@lamp ~/labAssignment$

Discussion

RSA is a cryptographic algorithm that employs a public key for encryption and a private key for decryption. This ensures that any information transmitted to the recipient is encrypted using the public key and can only be decrypted using the corresponding private key. In the given lab question, we were tasked with completing a code that performs the decryption process using the private key. In task 2 of the exam, we successfully decrypted the message and it revealed the content "Congratulation you solved it." Additionally, we utilized the same cipher to encrypt a new message and successfully decrypted it using our private key, as demonstrated in the output.