Operating System

EE 463

Assignment #1

Student: Mohammed Faisal Subhi

ID: 2035752

Q1: Write C program to find all possible combinations? And Write bash command to count them?

Source code

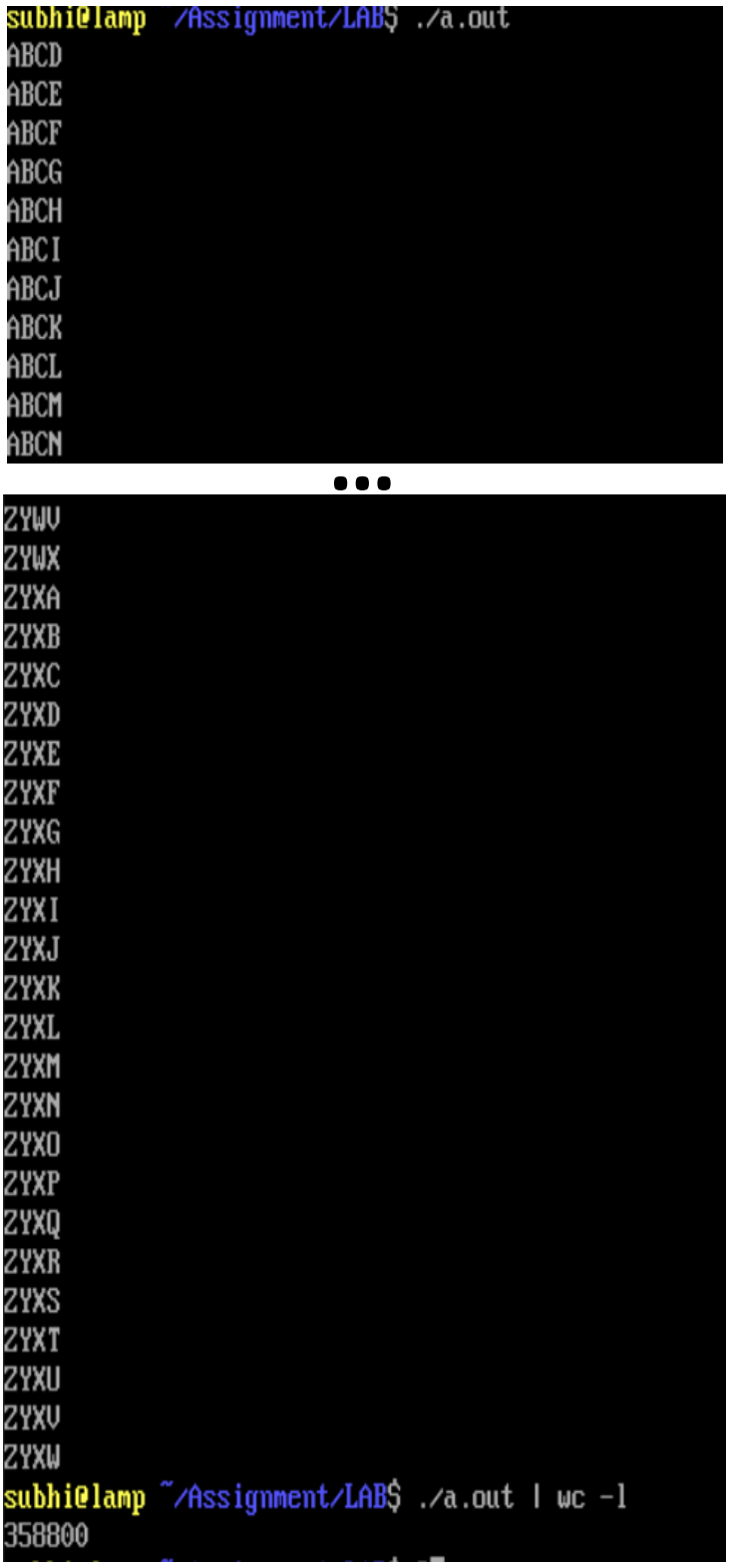
```c
// LAb Assignment 1
// Question 1
// Mohammed Faisal Subhi
// 2035752

#include <stdio.h>

// method that convert number into letter
char GetLetter(int number) {

 char Letter;

   switch (number) {
    case 1:
      Letter = 'A';
      break;
    case 2:
      Letter = 'B';
      break;
    case 3:
      Letter = 'C';
      break;
    case 4:
      Letter = 'D';
      break;
    case 5:
      Letter = 'E';
      break;
    case 6:
      Letter = 'F';
      break;
    case 7:
      Letter = 'G';
      break;
    case 8:
      Letter = 'H';
      break;
    case 9:
      Letter = 'I';
      break;
    case 10:
      Letter = 'J';
```

```
      break;
    case 11:
      Letter = 'K';
      break;
    case 12:
      Letter = 'L';
      break;
    case 13:
      Letter = 'M';
      break;
    case 14:
      Letter = 'N';
      break;
    case 15:
      Letter = 'O';
      break;
    case 16:
      Letter = 'P';
      break;
    case 17:
      Letter = 'Q';
      break;
    case 18:
      Letter = 'R';
      break;
    case 19:
      Letter = 'S';
      break;
    case 20:
      Letter = 'T';
      break;
    case 21:
      Letter = 'U';
      break;
    case 22:
      Letter = 'V';
      break;
    case 23:
      Letter = 'W';
      break;
    case 24:
      Letter = 'X';
      break;
    case 25:
      Letter = 'Y';
      break;
    case 26:
      Letter = 'Z';
```

```
      break;
  }
  return Letter;
}

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

char password[4];// to save the password
password[4] = '\0';


  for (int a = 1; a <= 26; a++) { // first Letter in the password

  // bring the Letter
   password[0] = GetLetter(a);

    for (int b= 1; b <= 26; b++) { // Second Letter in the password

    // check the condition
      if (b == a){
        continue;
      }

    // bring the Letter
    password[1] = GetLetter(b);

      for (int c = 1; c <= 26; c++) { // third Letter in the password

      // check the condition
      if (c == a){
        continue;
      }

      // check the condition
      if (c == b){
        continue;
      }

    // bring the Letter
    password[2] = GetLetter(c);

        for (int d = 1; d <= 26; d++) {

      // check the condition
      if (d == a){
        continue;
      }
      // check the condition
      if (d == b){
```

```
        continue;
      }
       // check the condition
      if (d == c){
        continue;
      }

    // bring the Letterss
    password[3] = GetLetter(d);

    // print the password
    printf("%c%c%c%c\n",
password[0],password[1],password[2],password[3]);
        }
      }
    }
  }
}
```

Screenshots



Q2:

Task1: Write C program to find the Private-Key by completing the given sample code decryptKey.c

Task2: Decrypt the following message:
858FF93C7C313EDC14E79A13EAF539D0893DACC7C70D335384965088E88AFC

Source code

```
// LAB Assignment 1
// Question 2
// Mohammed Faisal Subhi
// 2035752

/* *
* Author: Mohammed Subhi
* Description: This code could be use to decrept data
* RSA Decryption using OpenSSL library and Python encode/decode
* */

#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();

// Parameter
// Public-Key (Encryption Key)
char e[] = "010001";
// Product of large prime numbers
char n[] =
"E103ABD94892E3E74AFD724BF28E78366D9676BCCC70118BD0AA1968DBB143D1";
// totient of n, it is Euler's totient function phi(n)
char Qn[] =
"E103ABD94892E3E74AFD724BF28E78348D52298BD687C44DEB3A81065A7981A4";
```

```
// Here initialize all needed BIGNUM variables
// 1- Encryption Key variable
BIGNUM *EncryptionKey = BN_new();
// 2- Decryption Key variable
BIGNUM *DecryptionKey = BN_new();
// 3- product of large prime numbers p and q
BIGNUM *ProductPQ = BN_new();
// 4- Totient of (n) Euler's totient function
BIGNUM *Totient = BN_new();
// 5- Encrypted Message variable
BIGNUM *EncryptedMessage = BN_new();
// 6- Decrypted Ciphertext variable
BIGNUM *DecryptedCiphertext = BN_new();

// Find Decryption Key (d) using (e) and (Phin):
// 1- Assign value to (e) Encryption Key from hex
BN_hex2bn(&EncryptionKey, e);
// 2- Assign value to (Phin) Encryption Key from hex
BN_hex2bn(&Totient, Qn);
// 3- Calculate the Decryption Key (Private Key) d=e mod(Phi(n))
BN_mod_inverse(DecryptionKey, EncryptionKey, Totient, ctx);


char *CC= malloc(100 * sizeof(char));
printf("\nEnter your Encrypted Message:\n");
// Read the Encrypted Message from the user to variable CC
fgets(CC, 100, stdin);
// Assign the input value in variable (CC) to Encrypted Message
variable
BN_hex2bn(&EncryptedMessage, CC);


/*
Decrypt ciphertext using D=C^d(mod(n)) ,
where: (D) is the Decrypted Ciphertext and (C) is the Ciphertext
*/
// Assign value to (n) product of two large prime numbers from hex
BN_hex2bn(&ProductPQ, n);
// decrypt Ciphertext using the Private Key
BN_mod_exp(DecryptedCiphertext, EncryptedMessage, DecryptionKey,
ProductPQ, ctx);



// Convert Hex string to ASCII letters
printf("\nOriginal Message:\n");
char str1[500]="print(\"";
char *str2 = BN_bn2hex(DecryptedCiphertext);
```

```
char str3[]="\".decode(\"hex\"))";
strcat(str1,str2);
strcat(str1,str3);
char* args[]={"python2", "-c",str1, NULL};
execvp("python2", args);
return EXIT_SUCCESS;
}
```

Screenshots

```
subhi@lamp ~/Assignment/LAB$ ./a.out

Enter your Encrypted Message:
858FF93C7C313EDC14E79A13EAF539D0893DACC7C70D335384965088E88AFC

Original Message:
Congratulation you solved it.
```

```
subhi@lamp ~/Assignment/LAB$ ./encryptRSA

Enter Original Message:
Mohammed Faisal Subhi

Encoded Message:
4d6f68616d6d65642046616973616c205375626869

Re-enter Encoded Message:
4d6f68616d6d65642046616973616c205375626869

Encrypted Message:
7313611E0F428ACC680005B55F453FA8837465A76CC31E8F16E038069F24A232
subhi@lamp ~/Assignment/LAB$ ./a.out

Enter your Encrypted Message:
7313611E0F428ACC680005B55F453FA8837465A76CC31E8F16E038069F24A232

Original Message:
Mohammed Faisal Subhi
```

Discussion:

In RSA encryption a public key and private key are used. The public key can be published while the private key must be kept secret. The relation between the public key and the private key can be presented mathematically, using mathematics we can calculate the private key which is what we asked to do in the first task. As soon as you get the private key you can easily decrypt any Ciphertext (encrypted message) which is what we asked to do in the second task. Also using the private key, you can encrypt any Ciphertext and send it safely.