RSA Public-Key Encryption and Digital Signature Applications:

Task 1:

```
// Task1
/*
BN_CTX *ctx= BN_CTX_new();
BIGNUM *M= BN_new();
BIGNUM *e= BN_new();
BIGNUM *n= BN_new();
BIGNUM *encrypted= BN_new();
BIGNUM *decrypted= BN_new();
BIGNUM *d= BN_new();

BN_hex2bn(&M, "416361796970206769 7a6c6920626972206d6573616a21");
BN_hex2bn(&n, "BB300643E39AA365612115898C2737D969635148A40AAAD9F2A92E60A
7BB1BB7DA9A09F339FE02761FF451FF0FAFAFEA1C792D3C0114B2D4234FCFEABF1249C1");

BN_hex2bn(&e, "0D88C3");
BN_hex2bn(&d, "8D017DAF61EB9E6E08A74841F2F9B2F50D6913D605C98E416E06D8441DDBE
94F5F058E2FF8B629B59C98D4A6B799909455018CDE39C9FC3A4A74A6E483E45C07");

BN_mod_exp(encrypted,M,e,n,ctx);
printBN("Encrypted: ",encrypted);
BN_mod_exp(decrypted,encrypted,d,n,ctx);
printBN("Decrypted: ",decrypted);
*/
```

bn_sample.c

```
[06/25/22]seed@VM:~$ python3
Python 3.8.5 (default, Jul 28 2020, 12:59:40)
[GCC 9.3.0] on linux
Type "help", "copyright", "credits" or "license" for more information.
>>> import binascii
>>> x=b'Acayip gizli bir mesaj!'
>>> x=binascii.hexlify(x)
>>> x
b'416361796970206769 7a6c6920626972206d6573616a21'
>>>
[06/25/22]seed@VM:~$ gcc bn_sample.c -lcrypto
[06/25/22]seed@VM:~$ ./a.out
Encrypted:  2A3E2EEA928A52FF0A299493750EFDEE95394998D85E4484F68399222ECF515F1FC86813A19A2E
BEE1C802C45959254B3D2D9D4F502BD249B9D094E0A2B954B6
[06/25/22]seed@VM:~$ gcc bn_sample.c -lcrypto
[06/25/22]seed@VM:~$ ./a.out
Encrypted:  2A3E2EEA928A52FF0A299493750EFDEE95394998D85E4484F68399222ECF515F1FC86813A19A2E
BEE1C802C45959254B3D2D9D4F502BD249B9D094E0A2B954B6
Decrypted:  416361796970206769 7A6C6920626972206D6573616A21
```

Encrypted olan sonucu BN_mod_exp() fonksiyonunun içinde decrypt ederek. Mesajımızın hexadecimal sonucu ile doğrulamış olduk.

Task 2:

```c
//Task2
BN_CTX *ctx= BN_CTX_new();
BIGNUM *M= BN_new();
BIGNUM *e= BN_new();
BIGNUM *n= BN_new();
BIGNUM *encrypted= BN_new();
BIGNUM *decrypted= BN_new();
BIGNUM *d= BN_new();

BIGNUM *p= BN_new();
BIGNUM *q= BN_new();
BIGNUM *p1= BN_new();
BIGNUM *q1= BN_new();
BIGNUM *one= BN_new();
BIGNUM *N= BN_new();
BN_hex2bn(&p, "C353136B52414B12B4149F7FA641AE97A07C98292D4358227DFE0EA3BC4DAD7F");
BN_hex2bn(&q, "F555DEEF7084C34D2FB95C3B942BB4CCF06A8FD18CE63A87D63275CE06FE28BF");
BN_hex2bn(&e, "010001");
BN_hex2bn(&M, "427520646120696b696e63692067697a6c69206d6573616a");
BN_dec2bn(&one, "1"); //init 1 bignum and binary
BN_sub(p1,p,BN_value_one()); // p-1
BN_sub(q1,q,BN_value_one()); // q-1
BN_mul(N,p1,q1,ctx);//p-1 * q-1 = N
BN_mul(n,p,q,ctx);
printBN("Public key n: ",n);
BN_mod_inverse(d,e,N,ctx);// e * d mod N = 1
printBN("Derived key: ",d);
BN_mod_exp(encrypted,M,e,n,ctx);
printBN("Encrypted: ",encrypted);
BN_mod_exp(decrypted,encrypted,d,n,ctx);
printBN("Decrypted: ",decrypted);
```

```
[06/25/22]seed@VM:~$ python3
Python 3.8.5 (default, Jul 28 2020, 12:59:40)
[GCC 9.3.0] on linux
Type "help", "copyright", "credits" or "license" for more information.
>>> import binascii
>>> print(binascii.hexlify(b'Bu da ikinci gizli mesaj'))
b'427520646120696b696e63692067697a6c69206d6573616a'
>>>
[06/25/22]seed@VM:~$ gcc bn_sample.c -lcrypto
[06/25/22]seed@VM:~$ ./a.out
Public key n:  BB300643E39AA365612115898C2737D969635148A40AAAD9F2A92E60A7BB1BB7DA9
A09F339FE02761FF451FF0FAFAFEA1C792D3C0114B2D4234FCFEABF1249C1
Derived key:  89719CD812724451B73CECAC7F7D7873A9FF63FABB809DACC491D6DCDFABF2900C62
135A3F0795B111DF706E0453D70E23CA4713843836F89BFDB5941302C765
Encrypted:  85F35845F92B7FD091FF77E9AB80DF682E54A9868804AC28284652D766912FAF868BC7
55DD800F47CFC683EBF4BCD092C51366A7124F8E852B0DEF1ACE7169D7
Decrypted:  427520646120696B696E63692067697A6C69206D6573616A
[06/25/22]seed@VM:~$
```

Task 3:

```
//Task3
BN_CTX *ctx= BN_CTX_new();
BIGNUM *C= BN_new();
BIGNUM *n= BN_new();
BIGNUM *d= BN_new();
BIGNUM *decrypted= BN_new();
BN_hex2bn(&C,"7AA0FF25F5D5C94FBEA7109F8AA34A43ADA883EF30CE12A4595BBD92D36D91FBE43A8414003451770657265878827AB78549D6155500F9D319F892F8E74F07F");
BN_hex2bn(&n,"BB300643E39AA365612115898C2737D969635148A40AAAD9F2A92E60A7BB1BB7DA9A09F339FE02761FF451FF0FAFAFEA1C792D3C0114B2D4234FCFEABF1249C1");
BN_hex2bn(&d,"89719CD812724451B73CECAC7F7D7873A9FF63FABB809DACC491D6DCDFABF2900C62135A3F0795B111DF706E0453D70E23CA4713843836F89BFDB5941302C765");
BN_mod_exp(decrypted,C,d,n,ctx);
printBN("Decrypted: ",decrypted);
```

```
[06/25/22]seed@VM:~$ gcc bn_sample.c -lcrypto
[06/25/22]seed@VM:~$ ./a.out
Decrypted:  42756C64756D2067697A6C69206D6573616A6921
[06/25/22]seed@VM:~$ python3
Python 3.8.5 (default, Jul 28 2020, 12:59:40)
[GCC 9.3.0] on linux
Type "help", "copyright", "credits" or "license" for more information.
>>> print(bytes.fromhex('42756C64756D2067697A6C69206D6573616A6921').decode('utf-8'
))
Buldum gizli mesaji!
>>>
[06/25/22]seed@VM:~$
```

Task 4:

```
//Task4
BN_CTX *ctx= BN_CTX_new();
BIGNUM *M= BN_new();
BIGNUM *n= BN_new();
BIGNUM *sing= BN_new();
BIGNUM *d= BN_new();

BN_hex2bn(&M, "53616e612032206d696c796f6e206c69726120626f7263756d20766172");
BN_hex2bn(&n, "BB300643E39AA365612115898C2737D969635148A40AAAD9F2A92E60A7BB1BB7DA9A09F339FE02761FF451FF0FAFAFEA1C792D3C0114B2D4234FCFEABF1249C1");
BN_hex2bn(&d, "89719CD812724451B73CECAC7F7D7873A9FF63FABB809DACC491D6DCDFABF2900C62135A3F0795B111DF706E0453D70E23CA4713843836F89BFDB5941302C765");

BN_mod_exp(sing,M,d,n,ctx);
printBN("Signatured: ",sing);
```

```
[06/25/22]seed@VM:~$ python3
Python 3.8.5 (default, Jul 28 2020, 12:59:40)
[GCC 9.3.0] on linux
Type "help", "copyright", "credits" or "license" for more information.
>>> import binascii
>>> print(binascii.hexlify(b'Sana 1 milyon lira borcum var'))
b'53616e612031206d696c796f6e206c69726120626f7263756d20766172'
>>> print(binascii.hexlify(b'Sana 2 milyon lira borcum var'))
b'53616e612032206d696c796f6e206c69726120626f7263756d20766172'
```

```
[06/25/22]seed@VM:~$ gcc bn_sample.c -lcrypto
[06/25/22]seed@VM:~$ ./a.out
Signatured:  1E3E5D7D6B455B582F4B7D484EFD064636E18B3DAC3A21E739B89468E84AC4D812BE9
4FD2CD76FC68F649096D323CBB596062B154FD3401EB7EDDBDFB119C8BB
[06/25/22]seed@VM:~$ gcc bn_sample.c -lcrypto
[06/25/22]seed@VM:~$ ./a.out
Signatured:  18B1D1E4B0FA312F2010BD8801B873CB16562136D1C80DD30BF31CD273A6A1C0D6FA7
D31E8F12CFDB506FCD475A6D725D4EC3CB07E4F849C562AFA20A8EEC522
```

Görüldüğü üzere mesajlarda tek bit fark varken imzalar tamamen birbirinden farklı.

Task 5:

```c
//Task5
BN_CTX *ctx= BN_CTX_new();
BIGNUM *M= BN_new();
BIGNUM *M1= BN_new();
BIGNUM *M2= BN_new();
BIGNUM *n= BN_new();
BIGNUM *e= BN_new();
BIGNUM *sing1= BN_new();
BIGNUM *sing2= BN_new();


BN_hex2bn(&M, "4c61756e63682061206d697373696c652e");
BN_hex2bn(&n, "AE1CD4DC432798D933779FBD46C6E1247F0CF1233595113AA51B450F18116115");
BN_hex2bn(&sing1, "643D6F34902D9C7EC90CB0B2BCA36C47FA37165C0005CAB026C0542CBDB6802F");
BN_hex2bn(&sing2, "4E96B0012354774DD6C90215F0A51D356D08D9D64064C8703962C414378CE7F3");
BN_hex2bn(&e, "010001");
BN_mod_exp(M1,sing1,e,n,ctx);

if(BN_cmp(M1,M)==0)
        printf("S1 alice'e ait\n");
else
        printf("S1 alice'e ait degil\n");
BN_mod_exp(M2,sing2,e,n,ctx);
if(BN_cmp(M2,M)==0)
        printf("S2 alice'e ait\n");
else
        printf("S2 alice'e ait degil\n");
```

```
[06/25/22]seed@VM:~$ python3
Python 3.8.5 (default, Jul 28 2020, 12:59:40)
[GCC 9.3.0] on linux
Type "help", "copyright", "credits" or "license" for more information.
>>> import binascii
>>> print(binascii.hexlify(b'Launch a missile.'))
b'4c61756e63682061206d697373696c652e'
>>>
```

```
[06/25/22]seed@VM:~$ gcc bn_sample.c -lcrypto
[06/25/22]seed@VM:~$ ./a.out
S1 alice'e ait
S2 alice'e ait degil
```

İmzalar ile elde ettiğimiz mesajları, orijinal mesaj ile karşılaştırdığımızda S1'in alice'e ait olduğunu anladık

Crypto and Symmetric Key Encryption Applications:

Task 1:

```
YES, GOOD THEN...[06/25/22]seed@VM:~$ tr 'uocfdmvrkeganywbxlsjpz' 'HIDEARSTUOGNPFMWKYLCVB' < Bil420ciphertext.txt
HI DEAR STUDENT,
THIS IS THE HOMEWORK FOR THE COURSE OF INTRODUCTION TO CYBER SECUTITY.
TOBB ETU DEPARTMENT OF COMPUTER ENGINEERING IS THE FIRST AND ONLY DEPARTMENT FROM A TURKISH UNIVERSTY WITH INFORMATICS EU
ROPE MEMBERSHIP.
INFORMATICS EUROPE REPRESENTS THE ACADEMIC AND RESEARCH COMMUNITY IN INFORMATICS IN EUROPE AND NEIGHBOURING COUNTRIES.
IT AIMS TO IMPROVE, SHAPE AND REVIVE hUALITY IN RESEARCH, EDUCATION AND KNOWLEDGE TRANSFER IN INFORMATICS IN EUROPE BY BR
INGING TOGETHER UNIVERSITY DEPARTMENTS AND RESEARCH LABORATORIES AND CREATING A STRONG UNITY BETWEEN THEM.
BASED IN iURICH, SWITiERLAND, INFORMATICS EUROPE IS A NON-PROFIT COMMUNITY BASED ON MEMBERSHIP.
THE MAIN MISSION OF THE COMMUNITY IS TO PROMOTE RESEARCH, EDUCATION AND KNOWLEDGE TRANSFER IN INFORMATICS.
INFORMATICS EUROPE REPRESENTS OVER 120 UNIVERSITY DEPARTMENTS AND RESEARCH INSTITUTES ACROSS NEARLY 30 COUNTRIES IN EUROP
E.
INFORMATICS EUROPE MEMBERSHIP OF OUR DEPARTMENT WILL PLAY A CRUCIAL ROLE FOR OUR STUDENTS IN SHAPING THEIR ACADEMIC AND B
USINESS CAREERS INTERNATIONALLY.
DID YOU READ ALL?
YES, GOOD THEN...[06/25/22]seed@VM:~$ tr 'uocfdmvrkeganywbxlsjpzi' 'HIDEARSTUOGNPFMWKYLCVBZ' < Bil420ciphertext.txt
HI DEAR STUDENT,
THIS IS THE HOMEWORK FOR THE COURSE OF INTRODUCTION TO CYBER SECUTITY.
TOBB ETU DEPARTMENT OF COMPUTER ENGINEERING IS THE FIRST AND ONLY DEPARTMENT FROM A TURKISH UNIVERSTY WITH INFORMATICS EU
ROPE MEMBERSHIP.
INFORMATICS EUROPE REPRESENTS THE ACADEMIC AND RESEARCH COMMUNITY IN INFORMATICS IN EUROPE AND NEIGHBOURING COUNTRIES.
IT AIMS TO IMPROVE, SHAPE AND REVIVE hUALITY IN RESEARCH, EDUCATION AND KNOWLEDGE TRANSFER IN INFORMATICS IN EUROPE BY BR
INGING TOGETHER UNIVERSITY DEPARTMENTS AND RESEARCH LABORATORIES AND CREATING A STRONG UNITY BETWEEN THEM.
BASED IN ZURICH, SWITZERLAND, INFORMATICS EUROPE IS A NON-PROFIT COMMUNITY BASED ON MEMBERSHIP.
THE MAIN MISSION OF THE COMMUNITY IS TO PROMOTE RESEARCH, EDUCATION AND KNOWLEDGE TRANSFER IN INFORMATICS.
INFORMATICS EUROPE REPRESENTS OVER 120 UNIVERSITY DEPARTMENTS AND RESEARCH INSTITUTES ACROSS NEARLY 30 COUNTRIES IN EUROP
E.
INFORMATICS EUROPE MEMBERSHIP OF OUR DEPARTMENT WILL PLAY A CRUCIAL ROLE FOR OUR STUDENTS IN SHAPING THEIR ACADEMIC AND B
USINESS CAREERS INTERNATIONALLY.
DID YOU READ ALL?
```

İlk olarak frekans analizi sayfalarından harflerin frekansı ile İngilizce harflerinin frekansını değiştirerek denedim fakat doğru sonuç alamadım. Şifreli metine bakarken “uo cfdm vrkcfar,”

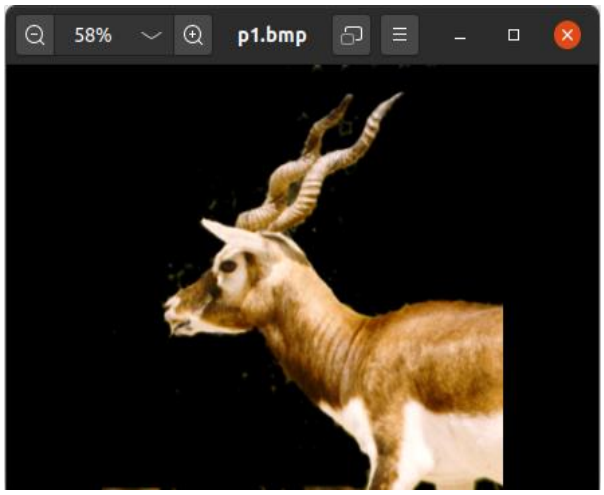
Başlangıç cümlesinin hi dear student olabilceğini deneyip tamamladım. Ayrıca iki harflı kelimelerin inglizcede en çok kullanan kelimeleri deniyerek doğru metni buldum.

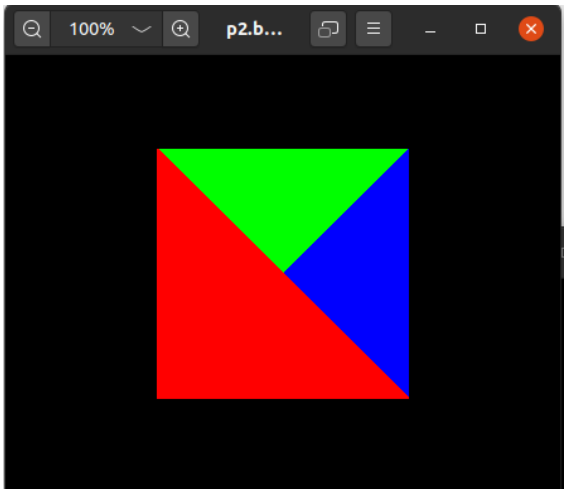
Task 2:

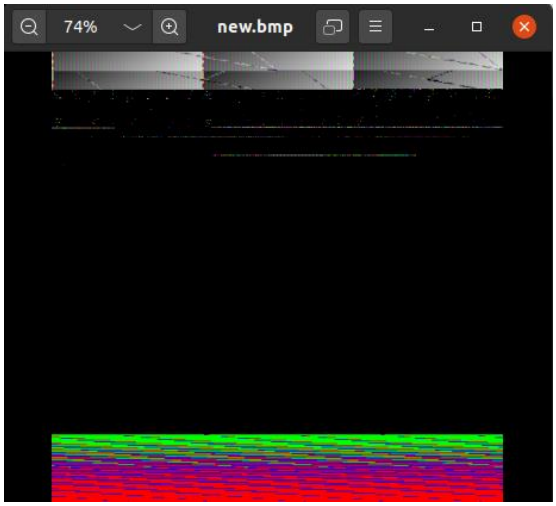
```
[06/25/22]seed@VM:~$ touch plain.txt
[06/25/22]seed@VM:~$ openssl aes-128-cbc -in plain.txt -out cipher.bin -K 001122
33445566778889aabbccddeeff -iv 0102030405060708
hex string is too short, padding with zero bytes to length
[06/25/22]seed@VM:~$ openssl aes-128-cfb -in plain.txt -out cipher.bin -K 001122
33445566778889aabbccddeeff -iv 0102030405060708
hex string is too short, padding with zero bytes to length
[06/25/22]seed@VM:~$ openssl bf-cbc -in plain.txt -out cipher.bin -K 00112233445
566778889aabbccddeeff -iv 0102030405060708
```

AES-128-cbc, AES-128-cfb ve bf-cbc şifrelemelerini denedim.

```
[06/25/22]seed@VM:~/Desktop$ head -c 54 p1.bmp > header
[06/25/22]seed@VM:~/Desktop$ tail -c +55 p2.bmp > body
[06/25/22]seed@VM:~/Desktop$ cat header body > new.bmp
```

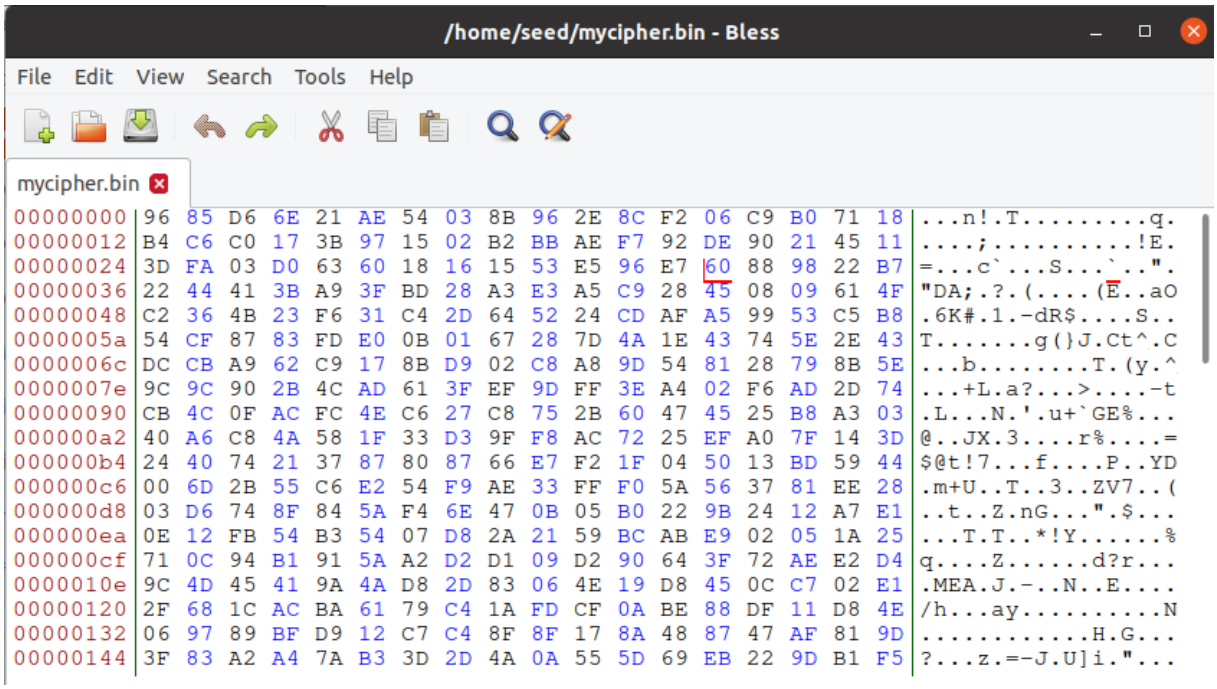


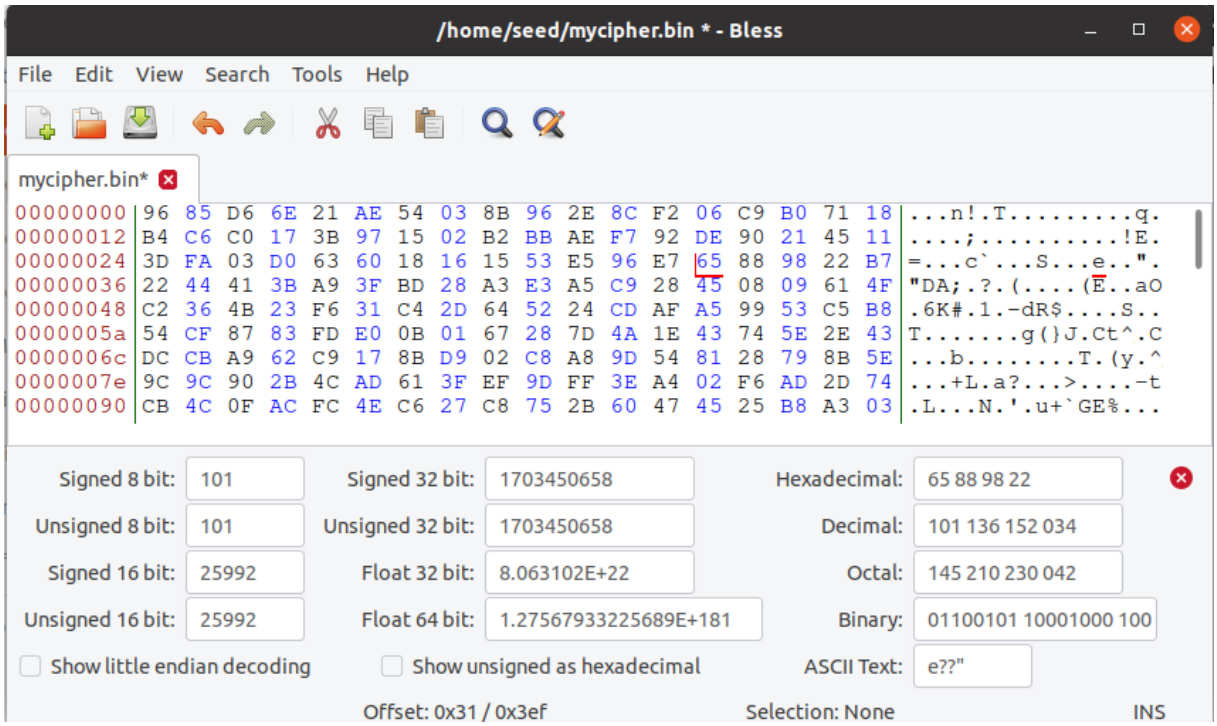




Görüldüğü üzere yeni oluşan new.bmp'den diğer fotoğraflar anlaşılmıyor.

Task 3:





50.byte'ı bozduk.

```
[06/25/22]seed@VM:~$ openssl aes-128-cbc -d -a -in mycipher.bin -out task3dec.ou
t -K 00112233445566778889aabbccddeeff -iv 0102030405060708
hex string is too short, padding with zero bytes to length
bad decrypt
139621023081792:error:0606506D:digital envelope routines:EVP_DecryptFinal_ex:wro
ng final block length:crypto/evp/evp_enc.c:572:
```

Decrypt ederken bozuk olduğu için edemedi.

EFB ve OFB bir önceki block'a bakmadığı için sadece 1 blok bozuldu. Fakat CBC ve CFB'de 2 block bozuldu.