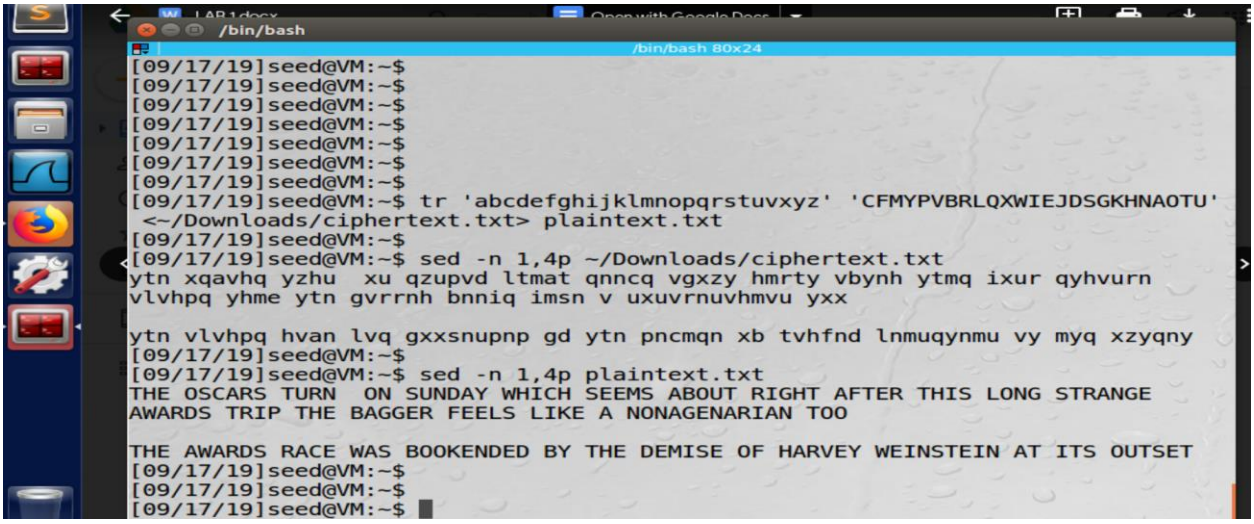
Task 1:

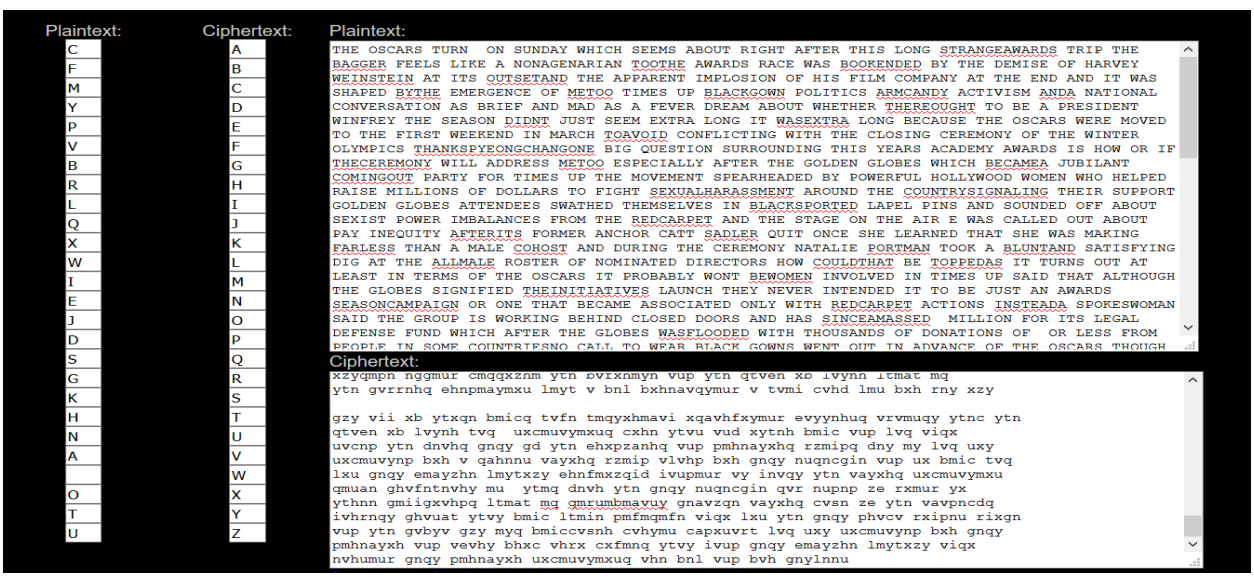
Steps: 1) find single works which can lead to "A" and "I",

Step2) use frequecy for top 3 letter

Step 3) used frequency for 2 and 3 letter words to find others

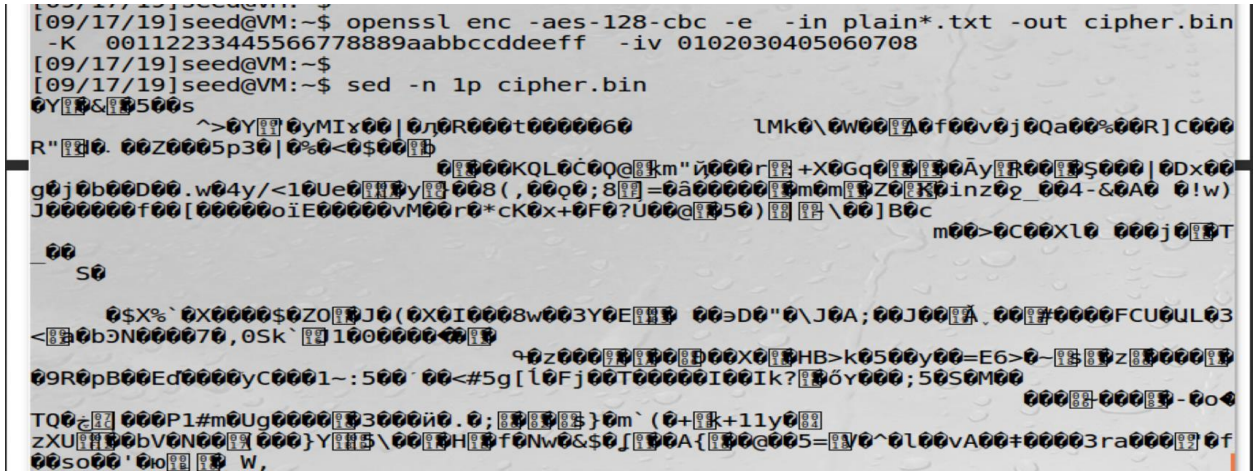
De-cypher and display first 4 line of the file



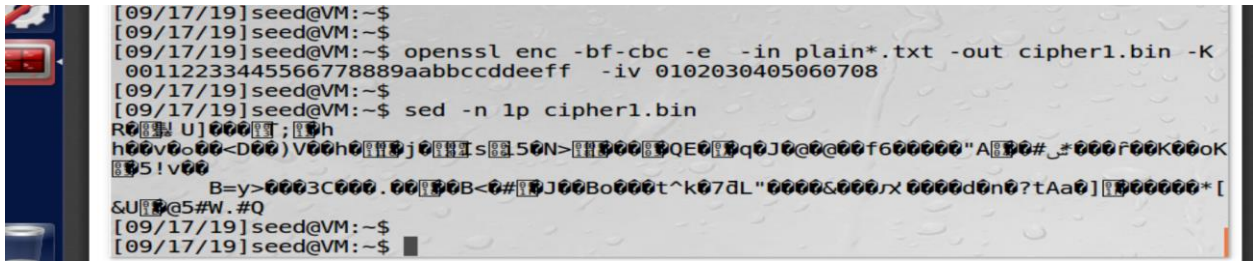


Task 2:

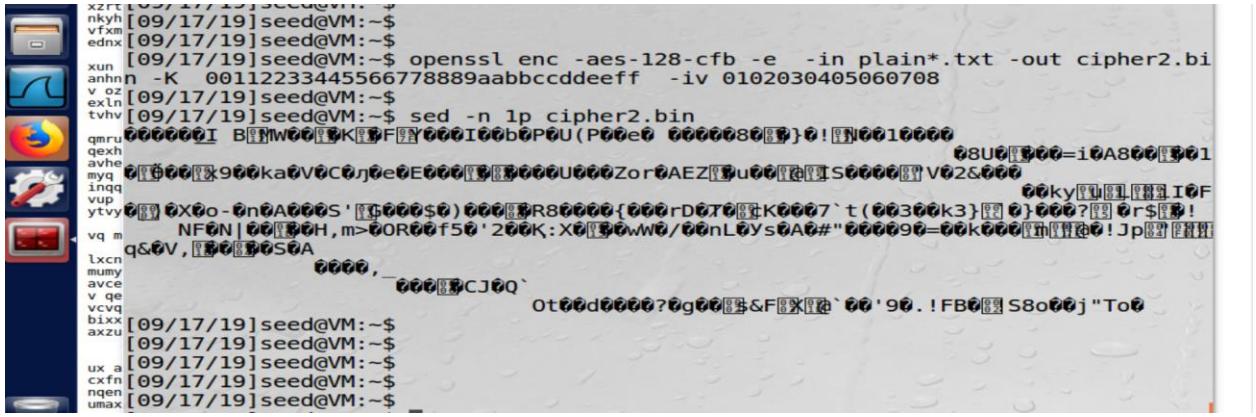
a)



b) -bf-cbc encryption



c) -aes-128-cfb



Task3:

1) Taken header part in "Header" and encrypted "full" with ecb and cbc

```
[09/19/19]seed@VM:~$ cp pic_original.bmp pic_original1.bmp
[09/19/19]seed@VM:~$
[09/19/19]seed@VM:~$ openssl enc -aes-128-cbc -e  -in pic_original.bmp -out cbc_
pic.bmp -K  00112233445566778889aabbccddeeff -iv 0102030405060708
[09/19/19]seed@VM:~$ openssl enc -des-ecb -e  -in pic_original.bmp -out ecb_pic.
bmp -K  00112233445566778889aabbccddeeff
hex string is too long
invalid hex key value
[09/19/19]seed@VM:~$ ls
android         Documents         header             pic_original.bmp  Videos
bin             Downloads         lab1               Pictures
cbc_pic.bmp     ecb_pic.bmp       lib                Public
Customization   examples.desktop  Music              source
Desktop         get-pip.py        pic_original1.bmp  Templates
[09/19/19]seed@VM:~$ openssl enc -des-ecb -e  -in pic_original.bmp -out ecb_pic.
bmp -K  0011223344556677
[09/19/19]seed@VM:~$
[09/19/19]seed@VM:~$
[09/19/19]seed@VM:~$ ls -lrt
total 2432
-rw-r--r-- 1 seed seed    8980 Jul 25  2017 examples.desktop
```

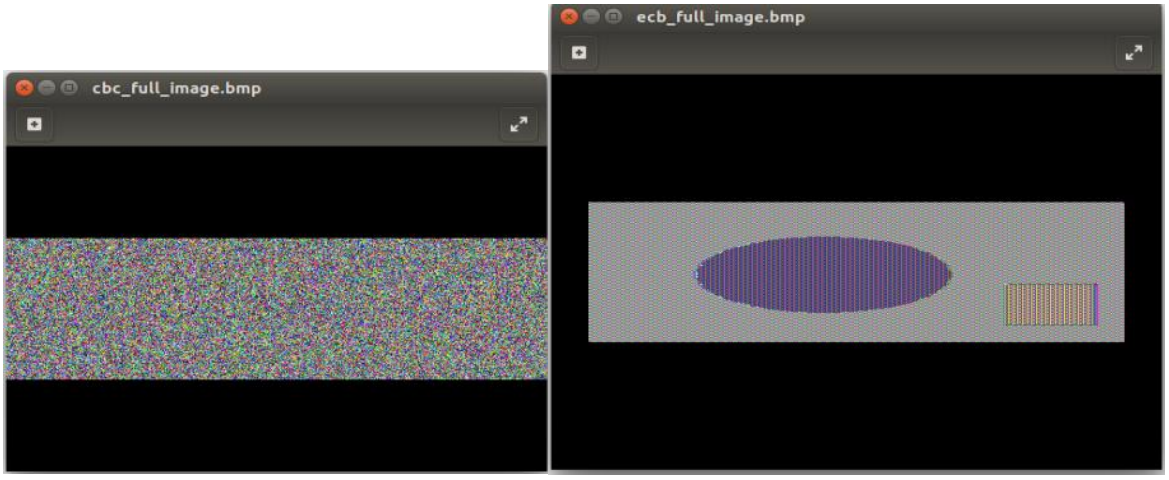
2)

Taking body out from both encryption and adding and appending with non-encrypted header

```
drwxrwxr-x 2 seed seed    4096 Sep 17 21:38 lab1
drwxr-xr-x 2 seed seed    4096 Sep 17 21:38 Downloads
-rw-rw-r-- 1 seed seed      54 Sep 17 21:39 header
-rw-rw-r-- 1 seed seed  184974 Sep 19 18:12 pic_original1.bmp
-rw-rw-r-- 1 seed seed  184976 Sep 19 18:18 cbc_pic.bmp
-rw-rw-r-- 1 seed seed  184976 Sep 19 18:23 ecb_pic.bmp
[09/19/19]seed@VM:~$
[09/19/19]seed@VM:~$
[09/19/19]seed@VM:~$
[09/19/19]seed@VM:~$
[09/19/19]seed@VM:~$ tail -c +55 ecb_pic.bmp > ecb_body
[09/19/19]seed@VM:~$ tail -c +55 cbc_pic.bmp > cbc_body
[09/19/19]seed@VM:~$ cat header ecb_body > ecb_full_image.bmp
[09/19/19]seed@VM:~$ cat header cbc_body > cbc_full_image.bmp
[09/19/19]seed@VM:~$
[09/19/19]seed@VM:~$
[09/19/19]seed@VM:~$
[09/19/19]seed@VM:~$
[09/19/19]seed@VM:~$
[09/19/19]seed@VM:~$
```

tool (already installed on our VM) to directly modify binary files.

Cbc image and ecb image respectively



Observation: Original image will have mostly 3 type of blocks, white pixel, red pixel, green pixel.

As CBC uses IV from previous block to encrypt next block so whole information seems to be random,

But on the other hand, ECB encrypt each bock independently, so all block (of same pixel) will be encrypted in same cypertext, for example most of red color block converted to pink color block.

Task 4: Padding

Created 3 files 5, 10, 15 byte

Encrypted them with cbc, ecb and decrypted to check padding. As we can see in below snapshot we have "0b" as padding till 16 byte block. While in 15byte file we have 1 byte padding

```
[09/19/19]seed@VM:~$
[09/19/19]seed@VM:~$
[09/19/19]seed@VM:~$ echo -n "12345" > 5_byte.txt
[09/19/19]seed@VM:~$ echo -n "1234567890" > 10_byte.txt
[09/19/19]seed@VM:~$ echo -n "123456789012345" > 15_byte.txt
[09/19/19]seed@VM:~$ openssl enc -aes-128-cbc -e  -in 5_byte.txt -out cbc_5_byte_enc -K  00112233445566778889aabbccdde
eff -iv 0102030405060708
[09/19/19]seed@VM:~$ openssl enc -aes-128-cbc -e  -in 15_byte.txt -out cbc_15_byte_enc -K  00112233445566778889aabbccd
deeff -iv 0102030405060708
[09/19/19]seed@VM:~$
[09/19/19]seed@VM:~$ hex
hex2hcd  hexdump
[09/19/19]seed@VM:~$ openssl enc -aes-128-cbc -d -nopad  -in cbc_5_byte_enc -out cbc_5_byte_dec -K  001122334455667788
89aabbccddeeff -iv 0102030405060708
[09/19/19]seed@VM:~$
[09/19/19]seed@VM:~$ xxd cbc_5_byte_dec
00000000: 3132 3334 350b 0b0b 0b0b 0b0b 0b0b 0b0b  12345...........
[09/19/19]seed@VM:~$
[09/19/19]seed@VM:~$
[09/19/19]seed@VM:~$ openssl enc -aes-128-cbc -d -nopad  -in cbc_10_byte_enc -out cbc_10_byte_dec -K  0011223344556677
8889aabbccddeeff -iv 0102030405060708
cbc_10_byte_enc: No such file or directory
3070965440:error:02001002:system library:fopen:No such file or directory:bss_file.c:398:fopen('cbc_10_byte_enc','r')
3070965440:error:20074002:BIO routines:FILE_CTRL:system lib:bss_file.c:400:
[09/19/19]seed@VM:~$ openssl enc -aes-128-cbc -d -nopad  -in cbc_15_byte_enc -out cbc_15_byte_dec -K  0011223344556677
8889aabbccddeeff -iv 0102030405060708
[09/19/19]seed@VM:~$
[09/19/19]seed@VM:~$
[09/19/19]seed@VM:~$
[09/19/19]seed@VM:~$ xxd cbc_15_byte_dec
00000000: 3132 3334 3536 3738 3930 3132 3334 3501  123456789012345.
[09/19/19]seed@VM:~$
```

```
[09/19/19]seed@VM:~$
[09/19/19]seed@VM:~$
[09/19/19]seed@VM:~$
[09/19/19]seed@VM:~$
[09/19/19]seed@VM:~$ openssl enc -aes-128-ecb -e  -in 5_byte.txt -out ecb_5_byte_enc -K  0011223344556677
[09/19/19]seed@VM:~$ openssl enc -aes-128-ecb -e  -in 15_byte.txt -out ecb_15_byte_enc -K  0011223344556677
[09/19/19]seed@VM:~$ openssl enc -aes-128-ecb -e  -in 10_byte.txt -out ecb_10_byte_enc -K  0011223344556677
[09/19/19]seed@VM:~$
[09/19/19]seed@VM:~$ openssl enc -aes-128-ecb -d -nopad  -in ecb_5_byte_enc -out ecb_5_byte_dec -K  0011223344556677
[09/19/19]seed@VM:~$ openssl enc -aes-128-ecb -d -nopad  -in ecb_10_byte_enc -out ecb_10_byte_dec -K  0011223344556677

[09/19/19]seed@VM:~$ openssl enc -aes-128-ecb -d -nopad  -in ecb_15_byte_enc -out ecb_15_byte_dec -K  0011223344556677

[09/19/19]seed@VM:~$
[09/19/19]seed@VM:~$ xxd ecb_5_byte_dec
00000000: 3132 3334 350b 0b0b 0b0b 0b0b 0b0b 0b0b  12345..........
[09/19/19]seed@VM:~$ xxd ecb_10_byte_dec
00000000: 3132 3334 3536 3738 3930 0606 0606 0606  1234567890......
[09/19/19]seed@VM:~$ xxd ecb_15_byte_dec
00000000: 3132 3334 3536 3738 3930 3132 3334 3501  123456789012345.
[09/19/19]seed@VM:~$
[09/19/19]seed@VM:~$
[09/19/19]seed@VM:~$
```

No visible padding in cfb and ofb

```
[09/19/19]seed@VM:~$
[09/19/19]seed@VM:~$ openssl enc -aes-128-cfb -e   -in 5_byte.txt -out cfb_5_byte_enc -K  00112233445566778889aabbccdd
eeff -iv 0102030405060708
[09/19/19]seed@VM:~$ openssl enc -aes-128-cfb -e   -in 15_byte.txt -out cfb_15_byte_enc -K  00112233445566778889aabbcc
ddeeff -iv 0102030405060708
[09/19/19]seed@VM:~$ openssl enc -aes-128-cfb -d  -nopad  -in cfb_5_byte_enc -out cfb_5_byte_dec -K  00112233445566778
889aabbccddeeff -iv 0102030405060708
[09/19/19]seed@VM:~$ openssl enc -aes-128-cfb -d  -nopad  -in cfb_15_byte_enc -out cfb_15_byte_dec -K  001122334455667
78889aabbccddeeff -iv 0102030405060708
[09/19/19]seed@VM:~$
[09/19/19]seed@VM:~$ xxd cfb_5_byte_dec
00000000: 3132 3334 35                             12345
[09/19/19]seed@VM:~$ xxd cfb_15_byte_dec
00000000: 3132 3334 3536 3738 3930 3132 3334 35    123456789012345
[09/19/19]seed@VM:~$
[09/19/19]seed@VM:~$
[09/19/19]seed@VM:~$ openssl enc -aes-128-ofb -e   -in 5_byte.txt -out ofb_5_byte_enc -K  00112233445566778889aabbccdd
eeff -iv 0102030405060708
[09/19/19]seed@VM:~$ openssl enc -aes-128-ofb -e   -in 15_byte.txt -out ofb_15_byte_enc -K  00112233445566778889aabbcc
ddeeff -iv 0102030405060708
[09/19/19]seed@VM:~$ openssl enc -aes-128-ofb -d  -nopad  -in ofb_5_byte_enc -out ofb_5_byte_dec -K  00112233445566778
889aabbccddeeff -iv 0102030405060708
[09/19/19]seed@VM:~$ openssl enc -aes-128-ofb -d  -nopad  -in ofb_15_byte_enc -out ofb_15_byte_dec -K  001122334455667
78889aabbccddeeff -iv 0102030405060708
[09/19/19]seed@VM:~$
[09/19/19]seed@VM:~$ xxd ofb_5_byte_dec
00000000: 3132 3334 35                             12345
[09/19/19]seed@VM:~$ xxd ofb_15_byte_dec
00000000: 3132 3334 3536 3738 3930 3132 3334 35    123456789012345
[09/19/19]seed@VM:~$
[09/19/19]seed@VM:~$
[09/19/19]seed@VM:~$
```

ECB and CBC used padding(block cypher 16byte block size) while CFB and OFB didn't(as they are streamcypher with 1 byte block size), it also supported by file size after decryption with padding

```
-rw-rw-r-- 1 seed seed        5 Sep 19 18:50 5_byte.txt
-rw-rw-r-- 1 seed seed       10 Sep 19 18:50 10_byte.txt
-rw-rw-r-- 1 seed seed       15 Sep 19 18:50 15_byte.txt
-rw-rw-r-- 1 seed seed       16 Sep 19 19:07 cbc_5_byte_enc
-rw-rw-r-- 1 seed seed       16 Sep 19 19:12 cbc_15_byte_enc
-rw-rw-r-- 1 seed seed       16 Sep 19 19:18 cbc_5_byte_dec
-rw-rw-r-- 1 seed seed       16 Sep 19 19:20 cbc_15_byte_dec
-rw-rw-r-- 1 seed seed       16 Sep 19 19:27 ecb_5_byte_enc
-rw-rw-r-- 1 seed seed       16 Sep 19 19:27 ecb_15_byte_enc
-rw-rw-r-- 1 seed seed       16 Sep 19 19:28 ecb_10_byte_enc
-rw-rw-r-- 1 seed seed       16 Sep 19 19:29 ecb_10_byte_dec
-rw-rw-r-- 1 seed seed       16 Sep 19 19:29 ecb_15_byte_dec
-rw-rw-r-- 1 seed seed        5 Sep 19 19:48 cfb_5_byte_enc
-rw-rw-r-- 1 seed seed       15 Sep 19 19:48 cfb_15_byte_enc
-rw-rw-r-- 1 seed seed        5 Sep 19 19:48 cfb_5_byte_dec
-rw-rw-r-- 1 seed seed       15 Sep 19 19:48 cfb_15_byte_dec
-rw-rw-r-- 1 seed seed        5 Sep 19 19:52 ofb_5_byte_enc
-rw-rw-r-- 1 seed seed       15 Sep 19 19:52 ofb_15_byte_enc
-rw-rw-r-- 1 seed seed        5 Sep 19 19:53 ofb_5_byte_dec
-rw-rw-r-- 1 seed seed       15 Sep 19 19:53 ofb_15_byte_dec
[09/19/19]seed@VM:~$
```

Task 5:

Create 1000byte file and encrypted them with ecb, cbc, ofc, ofc

```
[09/19/19]seed@VM:~$
[09/19/19]seed@VM:~$
[09/19/19]seed@VM:~$  openssl enc -aes-128-ofb -e  -in 1000*.txt -out ofb_1000byte_enc -K  00112233445566778889aabbccd
deeff -iv 0102030405060708
[09/19/19]seed@VM:~$ openssl enc -aes-128-cfb -e  -in 1000*.txt -out cfb_1000byte_enc -K  00112233445566778889aabbccdd
eeff -iv 0102030405060708
[09/19/19]seed@VM:~$ openssl enc -aes-128-cbc -e  -in 1000*.txt -out cbc_1000byte_enc -K  00112233445566778889aabbccdd
eeff -iv 0102030405060708
[09/19/19]seed@VM:~$ openssl enc -aes-128-ecb -e  -in 1000*.txt -out ecb_1000byte_enc -K  0011223344556677
[09/19/19]seed@VM:~$
[09/19/19]seed@VM:~$
[09/19/19]seed@VM:~$
[09/19/19]seed@VM:~$
[09/19/19]seed@VM:~$
[09/19/19]seed@VM:~$
[09/19/19]seed@VM:~$
[09/19/19]seed@VM:~$
[09/19/19]seed@VM:~$
[09/19/19]seed@VM:~$
[09/19/19]seed@VM:~$
[09/19/19]seed@VM:~$ ls -lrt 1000_byte.txt
-rw-rw-r-- 1 seed seed 1000 Sep 19 20:36 1000_byte.txt
[09/19/19]seed@VM:~$
[09/19/19]seed@VM:~$
```

Ofc:

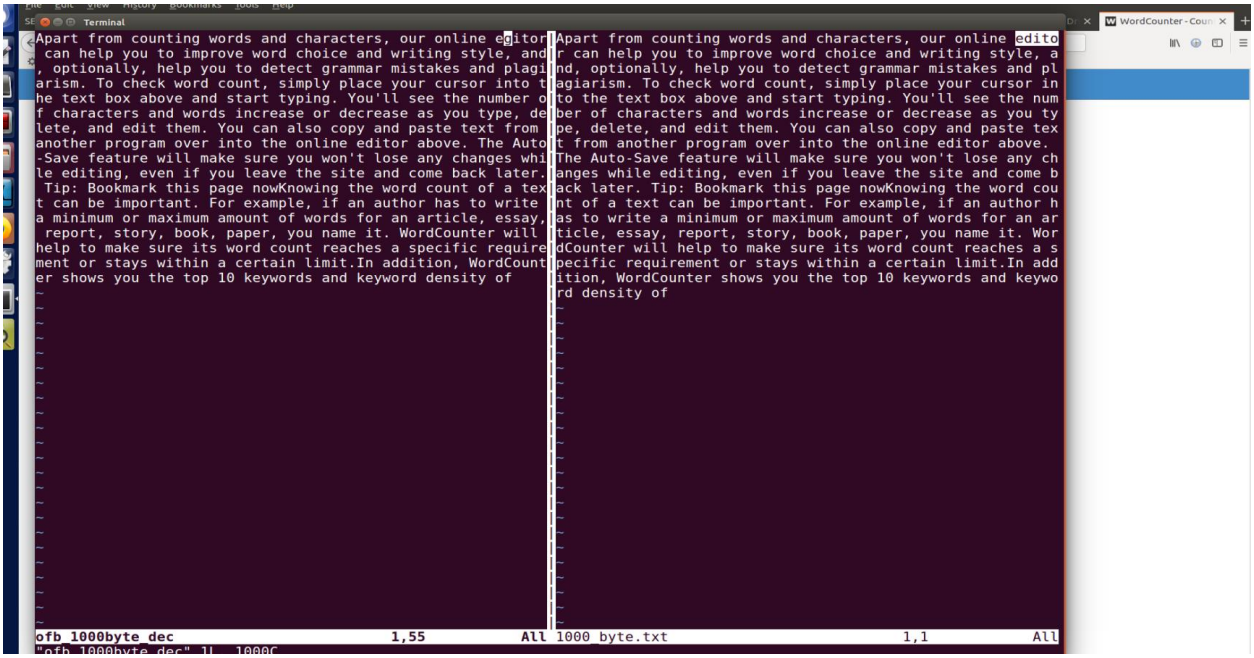
Uncorrupted encryption:

```
00000000 C6 F6 EE 57 B5 1F DC A3 C0 F3 9F 7E 1A 65 32 19 58 3A FE 27 CC 34 30 97 84 78 67 92 37 49 2B F7 B4 1F 5D 02 11 00 2D 26 39 86 04
0000002c 4F 1D C6 C6 62 4C DC FF 45 F3 81 4F 4F 1E 13 3A BC 1F 7B 74 15 17 B1 A0 F8 1E 2A 8D 46 DA 28 C3 DA AC BB 39 73 2E 05 DF BD 03 01
00000058 09 9F DF 09 02 7F 50 98 94 31 A6 F1 9E 8E 0A 69 1E 6A 8C 8F A0 E2 9E 4C 7C A5 74 BB 3C D5 13 9D 91 38 6F B3 D6 54 46 E6 6A 48 8B
00000084 87 BF 7A FC 21 FB 5B E3 7E F5 86 D0 BE 2A ED 6B 3C 26 33 85 E4 8B 22 E6 CA 38 10 84 94 F1 8C 1F 64 15 6D 01 52 3D CC 89 40 DA FA
000000b0 C8 F7 F1 80 C8 F3 90 B5 5C 2F 05 FC 08 E3 2D FA 6E 72 AF F4 14 2C B5 32 51 57 19 6C 9A 2E 9B 47 30 A7 D1 3E 34 1B A6 C8 3D F7 4D
000000dc 1A EA BA C1 48 E8 EB AD 0E F2 EC 40 FA 2A 7A A2 62 F9 A5 66 6A 80 3E 81 7F 94 32 75 8F E1 32 10 74 5A B9 C5 19 8C FF 33 AF 1A 0A
```

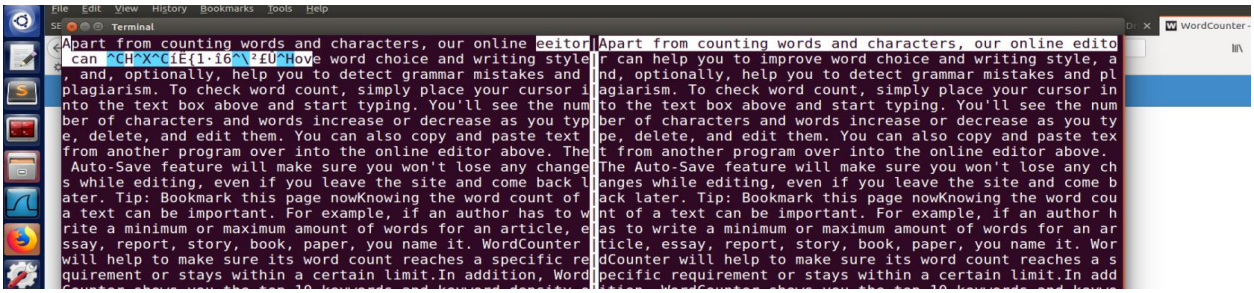
Corrupted one byte at 55th location:

```
ofb_1000byte_enc*
00000000 C6 F6 EE 57 B5 1F DC A3 C0 F3 9F 7E 1A 65 32 19 58 3A FE 27 CC 34 30 97 84 78 67 92 37 49 2B F7 B4 1F 5D 02 11 00 2D 26 39 86 04 F0 |...W.......~.e2.X:.'.40..xg.7I+...]
0000002c 4F 1D C6 C6 62 4C DC FF 45 F3 82 4F 4F 1E 13 3A BC 1F 7B 74 15 17 B1 A0 F8 1E 2A 8D 46 DA 28 C3 DA AC BB 39 73 2E 05 DF BD 03 01 3A |O...bL..E..OO..:..{t......*.F.(....
00000058 09 9F DF 09 02 7F 50 98 94 31 A6 F1 9E 8E 0A 69 1E 6A 8C 8F A0 E2 9E 4C 7C A5 74 BB 3C D5 13 9D 91 38 6F B3 D6 54 46 E6 6A 48 8B AE |......P..1.....i.j.....L|.t.<....8o
00000084 87 BF 7A FC 21 FB 5B E3 7E F5 86 D0 BE 2A ED 6B 3C 26 33 85 E4 8B 22 E6 CA 38 10 84 94 F1 8C 1F 64 15 6D 01 52 3D CC 89 40 DA FA 8F |..z.!.[.~....*.k<&3..."..8......d.m
000000b0 C8 F7 F1 80 C8 F3 90 B5 5C 2F 05 FC 08 E3 2D FA 6E 72 AF F4 14 2C B5 32 51 57 19 6C 9A 2E 9B 47 30 A7 D1 3E 34 1B A6 C8 3D F7 4D B9 |........\/....-.nr...,.2QW.l...G0..
000000dc 1A EA BA C1 48 E8 EB AD 0E F2 EC 40 FA 2A 7A A2 62 F9 A5 66 6A 80 3E 81 7F 94 32 75 8F E1 32 10 74 5A B9 C5 19 8C FF 33 AF 1A 0A 4B |....H......@.*z.b..fj.>...2u..2.tZ.
00000108 EA 7A EB D8 5F 43 79 67 EC C9 08 70 30 20 5F 4E 55 F8 30 27 67 4D F1 FE EF F7 E3 00 94 B6 7E 04 29 D4 F2 76 35 28 8B BD 12 CB 49 F2 |.z.._Cyg...p0 _NU.0'gM........~.)..
```

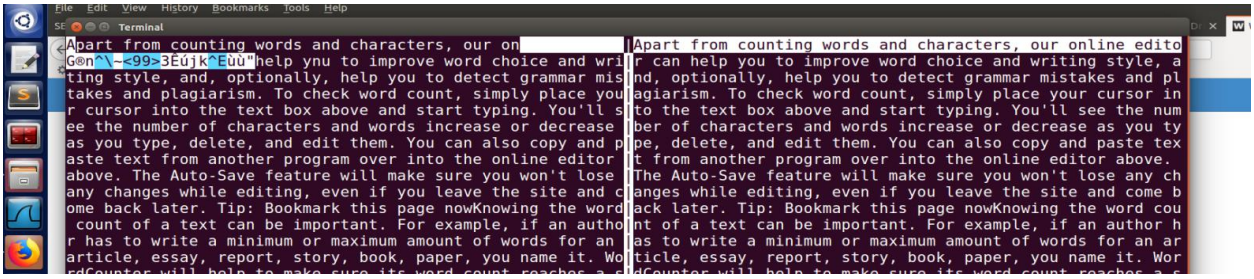
Decrypted and compared: “d” in original became “g”; Only one bit is corrupted in plaint text.



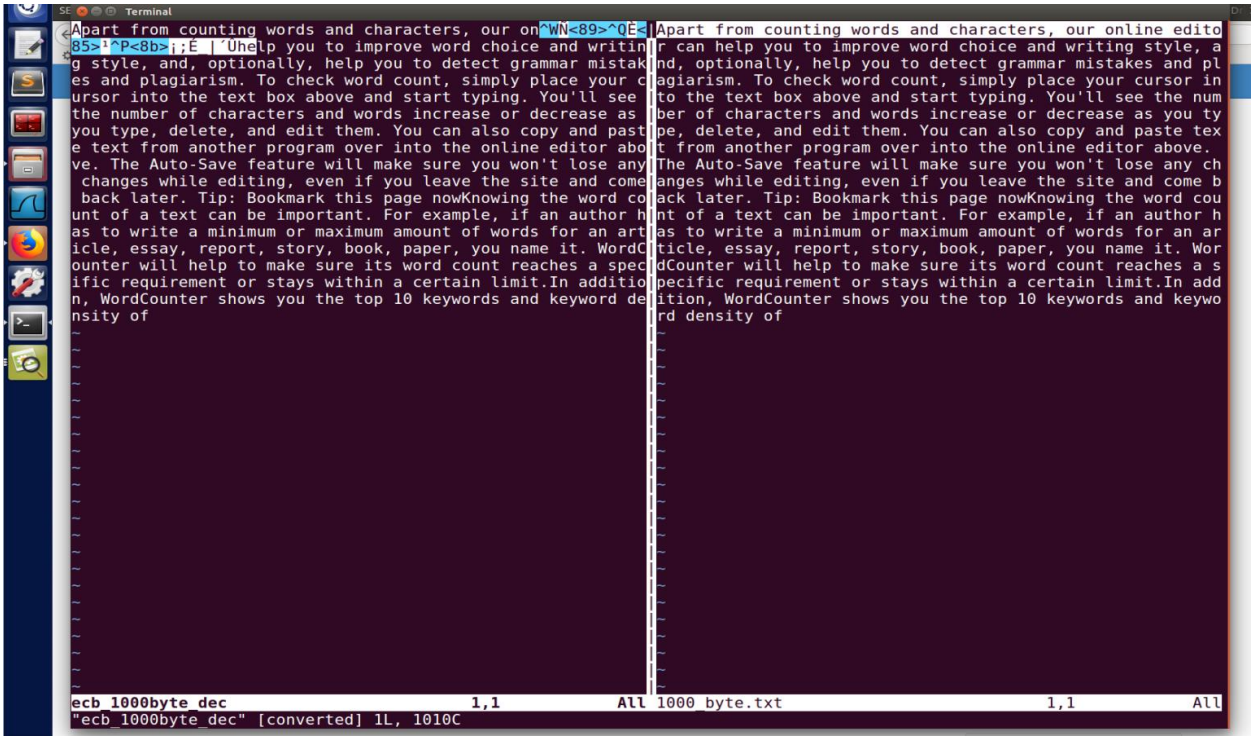
CFB: on changing 1 byte. Whole block got corrupted



CBC: whole block is corrupted



ECB: one block(16byte) corrupted



Task 7: Next page

Task 7:

```
Terminal
#include<stdio.h>
#include<stdlib.h>
#include<string.h>
#include<openssl/evp.h>
#include<ctype.h>
int main(){
        FILE *wfptr = NULL;
        ssize_t readlen=0;
        size_t len;
        char padd[16] ="###############";
        char *word = (char*) malloc(1024);
        int tmplen =0;
        unsigned char outbuf[1024 + EVP_MAX_BLOCK_LENGTH];
        EVP_CIPHER_CTX ctx;
        char inText[] = "This is a top secret.";
        unsigned char iv[] = {0xaa,0xbb,0xcc,0xdd,0xee,0xff,0x00,0x99,0x88,0x77,0x66,0x55,0x44,0x33,0x22,0x11};
        unsigned char cypher_text[] = "062ff0112cb32d04d0adcfa02d215abd40a5f932da1ebbd3744de5d16be5a4d7";
        wfptr = fopen("words.txt","r");
        if(wfptr == NULL) {
                printf("couldnt open password file");
                return 0;
        }
        int count =0,outlen = 0;
        while((readlen = getline(&word, &len, wfptr)!=-1)){
                outlen=0,tmplen=0;
                printf("\n %s  ",word);
                word[strlen(word)-1] ='\0';
                if(strlen(word)<16) {
                        strncat(word,padd,16-strlen(word));
                        word[strlen(word)] = '\0';
                }
                EVP_CIPHER_CTX_init(&ctx);
                EVP_EncryptInit_ex(&ctx, EVP_aes_128_cbc(), NULL, word, iv);
                if(!EVP_EncryptUpdate(&ctx, outbuf, &outlen, inText, strlen(inText)))
                {
                        /* Error */
                        EVP_CIPHER_CTX_cleanup(&ctx);
                        return 0;
                }
                if(!EVP_EncryptFinal_ex(&ctx,outbuf+ outlen, &tmplen)){
                        EVP_CIPHER_CTX_cleanup(&ctx);
                }
                outlen +=tmplen;
                char* enc_str = (char*) malloc(12*outlen +1);
"find_key.c" 57L, 1585C                                                    1,1           Top
```

```
                char* enc_ptr = enc_str;
                for(int i=0;i<outlen;i++) {
                        enc_str +=sprintf( enc_str,"%02x",outbuf[i]);
                }
                *(enc_str + 1) = '\0';
                printf("\n%s\n", enc_ptr);
                EVP_CIPHER_CTX_cleanup(&ctx);

                free(enc_ptr);
                if(!strcmp(enc_ptr,cypher_text)){
                        printf("Found the match");
                        break;
                }
        }
        free(word);
        fclose(wfptr);
        return 0;
}
"find_key.c" 62L, 1668C written                                        54,20-34      Bot
```

Running

```
[09/20/19]seed@VM:~$
[09/20/19]seed@VM:~$
[09/20/19]seed@VM:~$
[09/20/19]seed@VM:~$ vim find_key.c
[09/20/19]seed@VM:~$ gcc find_key.c -lcrypto
[09/20/19]seed@VM:~$
[09/20/19]seed@VM:~$ rm 2
[09/20/19]seed@VM:~$ ./a.out > 2
[09/20/19]seed@VM:~$ vim 2
```

Key used for encryption: “secret”