

Task 1:

- Steps: 1) find single words which can lead to "A" and "I",
- Step2) use frequency 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

```
seed@VM:~$  
[09/17/19]seed@VM:~$  
[09/17/19]seed@VM:~$  
[09/17/19]seed@VM:~$  
[09/17/19]seed@VM:~$  
[09/17/19]seed@VM:~$  
[09/17/19]seed@VM:~$  
[09/17/19]seed@VM:~$  
[09/17/19]seed@VM:~$ tr 'abcdefghijklmnopqrstuvwxyz' 'CFMYPVBRLQXWIEJDSGKHNAOTU'  
<~/Downloads/ciphertext.txt> plaintext.txt  
[09/17/19]seed@VM:~$  
[09/17/19]seed@VM:~$ sed -n 1,4p ~/Downloads/ciphertext.txt  
ytn xqavhq yzhu xu qzupvd ltmat qnncq vxgzy hmrt ybnyh ytmq ixur qyhvurn  
vlvhpq yhme ytn gvrnrh bnaiq imsn v uxuvrnvuhmvu yxx  
  
ytn vlvhpq hvan lvq gxxsnupnp gd ytn pncmqn xb tvhfnd lnmuqynmu vy myq xzyqny  
[09/17/19]seed@VM:~$  
[09/17/19]seed@VM:~$ sed -n 1,4p plaintext.txt  
THE OSCARS TURN ON SUNDAY WHICH SEEMS ABOUT RIGHT AFTER THIS LONG STRANGE  
AWARDS TRIP THE BAGGER FEELS LIKE A NONAGENARIAN TOO  
  
THE AWARDS RACE WAS BOOKENDED BY THE DEMISE OF HARVEY WEINSTEIN AT ITS OUTSET  
[09/17/19]seed@VM:~$  
[09/17/19]seed@VM:~$  
[09/17/19]seed@VM:~$
```

Plaintext:	Ciphertext:	Plaintext:
C	A	THE OSCARS TURN ON SUNDAY WHICH SEEMS ABOUT RIGHT AFTER THIS LONG STRANGE AWARDS TRIP THE
F	B	BAGGER FEELS LIKE A NONAGENARIAN TOO THE AWARDS RACE WAS BOOKENDED BY THE DEMISE OF HARVEY
M	C	WEINSTEIN AT ITS OUTSET AND THE APPARENT IMPLSION OF HIS FILM COMPANY AT THE END AND IT WAS
Y	D	SHAPED BY THE EMERGENCE OF METOO TIMES UP BLACKGOWN POLITICS ARMCANDY ACTIVISM AND A NATIONAL
P	E	CONVERSATION AS BRIEF AND MAD AS A FEVER DREAM ABOUT WHETHER THERE COULD TO BE A PRESIDENT
V	F	WINFREY THE SEASON DIDNT JUST SEEM EXTRA LONG IT WAS EXTRA LONG BECAUSE THE OSCARS WERE MOVED
B	G	TO THE FIRST WEEKEND IN MARCH TO AVOID CONFLICTING WITH THE CLOSING CEREMONY OF THE WINTER
R	H	OLYMPICS THANKSPYEONGCHANGONE BIG QUESTION SURROUNDING THIS YEARS ACADEMY AWARDS IS HOW OR IF
L	I	THE CEREMONY WILL ADDRESS METOO ESPECIALLY AFTER THE GOLDEN GLOBES WHICH BECAME A JUBILANT
Q	J	COMING OUT PARTY FOR TIMES UP THE MOVEMENT SPEARHEADED BY POWERFUL HOLLYWOOD WOMEN WHO HELPED
X	K	RAISE MILLIONS OF DOLLARS TO FIGHT SEXUAL HARASSMENT AROUND THE COUNTRY SIGNALING THEIR SUPPORT
W	L	GOLDEN GLOBES ATTENDEES SWATHED THEMSELVES IN BLACKSPORED LAPEL PINS AND SOUNDED OFF ABOUT
I	M	SEXIST POWER IMBALANCES FROM THE RED CARPET AND THE STAGE ON THE AIR E WAS CALLED OUT ABOUT
E	N	PAY INEQUITY AFTER ITS FORMER ANCHOR CATT SADLER QUIT ONCE SHE LEARNED THAT SHE WAS MAKING
J	O	FARLESS THAN A MALE COHOST AND DURING THE CEREMONY NATALIE PORTMAN TOOK A BLUNT AND SATISFYING
D	P	DIG AT THE ALL MALE ROSTER OF NOMINATED DIRECTORS HOW COULD THAT BE TOPPED AS IT TURNS OUT AT
S	Q	LEAST IN TERMS OF THE OSCARS IT PROBABLY WONT BE WOMEN INVOLVED IN TIMES UP SAID THAT ALTHOUGH
G	R	THE GLOBES SIGNIFIED THE INITIATIVES LAUNCH THEY NEVER INTENDED IT TO BE JUST AN AWARDS
K	S	SEASON CAMPAIGN OR ONE THAT BECAME ASSOCIATED ONLY WITH RED CARPET ACTIONS INSTEAD A SPOKESWOMAN
H	T	SAID THE GROUP IS WORKING BEHIND CLOSED DOORS AND HAS SINCE AMASSED MILLION FOR ITS LEGAL
N	U	DEFENSE FUND WHICH AFTER THE GLOBES WAS FLOODED WITH THOUSANDS OF DONATIONS OF OR LESS FROM
A	V	PEOPLE IN SOME COUNTRIES ENO CALL TO WEAR BLACK GOWNS WENT OUT IN ADVANCE OF THE OSCARS THOUGH
O	W	Ciphertext:
T	X	xzyqpn nggmur cmqzxnm ytn evxnmyn vup ytn qtven xb lvyhn ltmat mq
U	Y	ytn gvrnrh q ehnpmaymxu lmyt v bnl bxhnavqymur v tvmi cvhd lmu bxh rny xzy
	Z	gzy vii xb ytxqn bmicq tvfn tmqyxhmavi xqavhfxymur evyynhuq vrvmuqy ytn ytn
		qtven xb lvyhn tvq uxcmuvmxux cxhn ytvu vud xytnh bmic vup lvq viqx
		uvcnp ytn dnvhq gnqy gd ytn ehxpzanhq vup pmhnayxhq rzmlpq dny my lvq uxy
		uxcmuvynp bxh v qahnnu vayxhq rzmlp vlvhp bxh gnqy nuqncgin vup ux bmic tvq
		lxu gnqy emayzhn lmytxzy ehnmzxzid ivupmur vy invqy ytn vayxhq uxcmuvmxux
		qmuau ghvntvny mu ytmq dnvb ytn gnqy nuqncgin qvi nupnp ze xzmur yx
		ythnn gmiigxvhpq ltmat mq gmrumbmauy gnayzqn vayxhq cvsn ze ytn vavpncdq
		ivhrnqy ghvuat ytv y bmic ltmin pmfmqmfn viqx lxu ytn gnqy phvce rxipnu rixgn
		vup ytn gbyvy gzy myq bmiccvsnh cvhymu capxuvrt lvq uxy uxcmuvmynp bxh gnqy
		pmhnayxh vup vevhy bxhc vxrx cxfmnq ytv y ivup gnqy emayzhn lmytxzy viqx
		nvhumur gnqy pmhnayxh uxcmuvmxux vhn bnl vup bvh gnylnnu

Task 2

a)

```
[09/17/19]seed@VM:~$ openssl enc -aes-128-cbc -e -in plain*.txt -out cipher.bin
-K 00112233445566778889aabbccddeeff -iv 0102030405060708
[09/17/19]seed@VM:~$
[09/17/19]seed@VM:~$ sed -n 1p cipher.bin
0Y[0]S[0]500s
^>0Y[0]0yMIx00|0лR000t0000060          LMk0\0w00[0]A0f00v0j0Qa00%00R]C000
R"[0]0. 00Z0005p30|0%0<0$00[0]b
0[0]00KQL0C0Q@0km"й000r[0]+X0Gq0[0]000Aÿ[0]R00[0]S000|0Dx00
g0j0b00D00.w04y/<10Ue0[0]0y[0]-008(,00q0;8[0]=0â00000[0]m0m[0]Z0[0]K0inz0p_004-&0A0_0!w)
J000000f00[00000oIE00000vM00r0*cK0x+0F0?U00@0[0]50)[0][0]\00]B0c
m00>0C00Xl0_000j0[0]T
00
S0
0$X%`0X0000$0Z0[0]J0(0X0I0008w003Y0E[0]0 00D0"0\J0A;00J00[0]A_00[0]#0000FCU0UL03
<[0]b0N0000070,0Sk`[0]10000000[0]
0z000[0]0[0]00000X0[0]HB>k0500y00=E6>0-[0]0[0]z0[0]000[0]
09R0pB00Ed0000yC0001~:500`00<#5g[10Fj00T00000I00Ik?0[0]ôY000;50S0M00
000[0]0-0o0
T00[0]000P1#m0Ug0000[0]03000й.0;[0]0[0]0[0]0m` (0+[0]k+11y0[0]
zXU[0]0bV0N00[0]00[0]Y[0]0[0]0\00[0]0H[0]0f0Nw0&$0[0]0[0]A{0[0]0@005=[0]0^0l00vA00#00003ra000[0]0f
00so00'0io[0]0[0]W,
```

b) -bf-cbc encryption

```
[09/17/19]seed@VM:~$
[09/17/19]seed@VM:~$
[09/17/19]seed@VM:~$ openssl enc -bf-cbc -e -in plain*.txt -out cipher1.bin -K
00112233445566778889aabbccddeeff -iv 0102030405060708
[09/17/19]seed@VM:~$
[09/17/19]seed@VM:~$ sed -n 1p cipher1.bin
R0[0]U[0]000[0];[0]h
h00v0o00<D00)V00h0[0]j0[0]s0[0]50N>[0]000[0]QE0[0]q0J0@0@00f600000"A[0]0#_#000f00K00oK
[0]5!v00
B=y>0003C000.00[0]0B<0#[0]J00Bo000t^k07dL"0000&000rx0000d0n0?tAa0[0]000000*[
&U[0]0@5#W.#Q
[09/17/19]seed@VM:~$
[09/17/19]seed@VM:~$
```

c) -aes-128-cfb

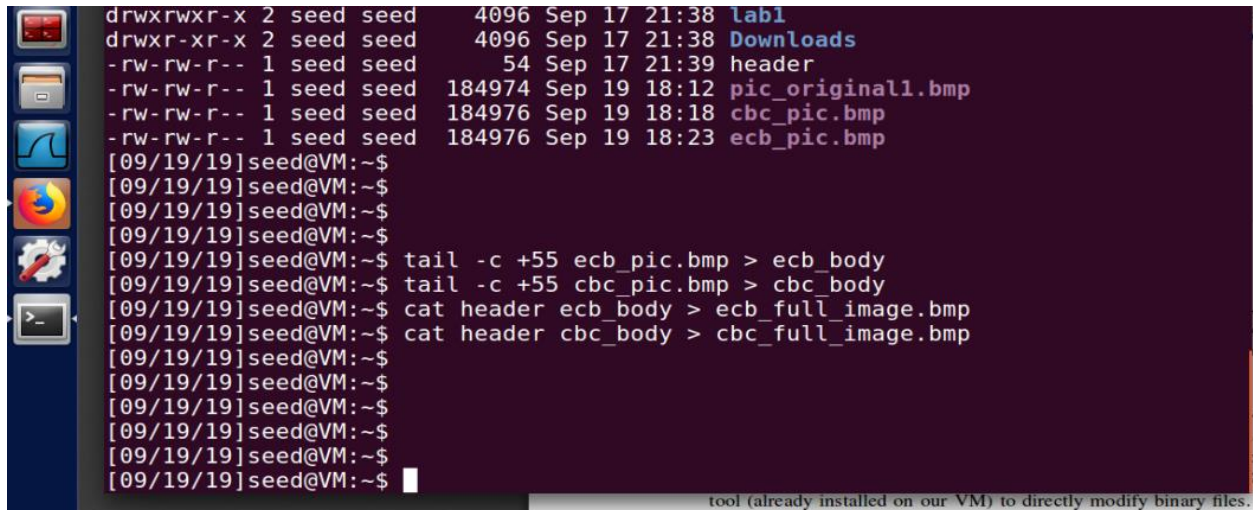
```
xzrt [09/17/19]seed@VM:~$
nkyh [09/17/19]seed@VM:~$
vfxm [09/17/19]seed@VM:~$
ednx [09/17/19]seed@VM:~$
xun [09/17/19]seed@VM:~$ openssl enc -aes-128-cfb -e -in plain*.txt -out cipher2.bi
anhn -K 00112233445566778889aabbccddeeff -iv 0102030405060708
v_oz [09/17/19]seed@VM:~$
exln [09/17/19]seed@VM:~$
tvhv [09/17/19]seed@VM:~$ sed -n 1p cipher2.bin
000000I B[0]W00[0]K[0]F[0]000I00b0P0U(P00e0_0000080[0])!N0010000
08U0[0]00=i0A800[0]01
0[0]00[0]X900ka0V0C0л0e0E000[0]0[0]000U000Zor0AEZ[0]u00[0]0[0]S0000[0]V02&000
00ky[0]0[0]V00I0F
0[0]X0o-0n0A000S'[0]000$0)000[0]R80000{000rD070[0]K0007` t(00300k3}[0]0)000?0[0]0r$0[0]!
NF0N|00[0]00H,m>0OR00f50'200K:X0[0]0wW0/00nL0ys0A0#"000090=00k000[0]h[0]0[0]0!Jp[0]0[0]0[0]
q&0V,[0]0[0]0S0A
0000,
000[0]CJ0Q`
Ot00d0000?0g00[0]0&F[0]X[0]00`00'90.!FB0[0]S8o00j"To0
[09/17/19]seed@VM:~$
[09/17/19]seed@VM:~$
[09/17/19]seed@VM:~$
[09/17/19]seed@VM:~$
[09/17/19]seed@VM:~$
```


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
```

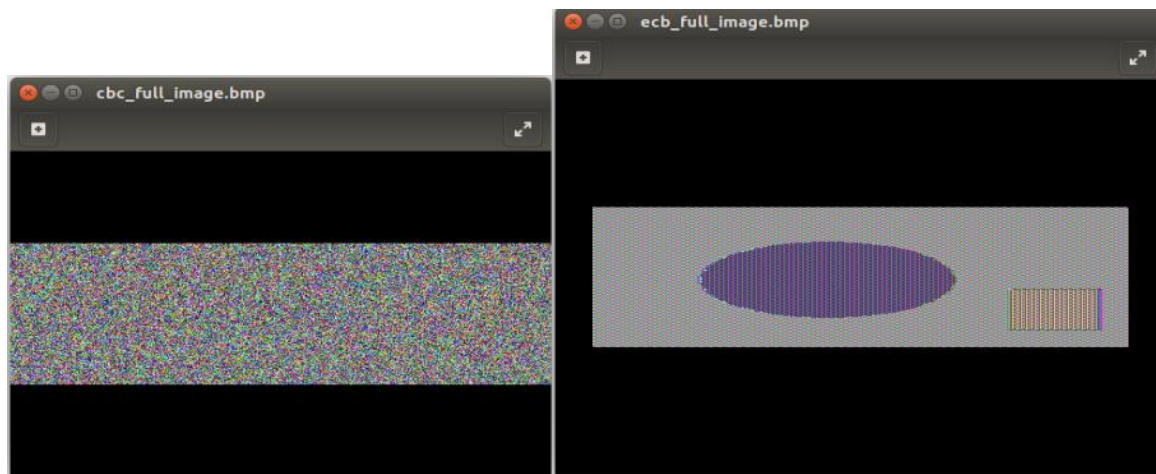
Taking body out from both encryption and adding and appending with non-encrypted header



```
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 block independently, so all block (of same pixel) will be encrypted in same ciphertext, 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 15 byte 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 00112233445566778889aabbccddeeff -iv 0102030405060708
[09/19/19]seed@VM:~$ openssl enc -aes-128-cbc -e -in 15_byte.txt -out cbc_15_byte_enc -K 00112233445566778889aabbccddeeff -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 00112233445566778889aabbccddeeff -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 00112233445566778889aabbccddeeff -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:BIIO 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 00112233445566778889aabbccddeeff -iv 0102030405060708
[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:~$ 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 00112233445566778889aabbccddeeff -iv 0102030405060708
[09/19/19]seed@VM:~$ openssl enc -aes-128-cfb -e -in 15_byte.txt -out cfb_15_byte_enc -K 00112233445566778889aabbccddeeff -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 00112233445566778889aabbccddeeff -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 00112233445566778889aabbccddeeff -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:~$ openssl enc -aes-128-ofb -e -in 5_byte.txt -out ofb_5_byte_enc -K 00112233445566778889aabbccddeeff -iv 0102030405060708
[09/19/19]seed@VM:~$ openssl enc -aes-128-ofb -e -in 15_byte.txt -out ofb_15_byte_enc -K 00112233445566778889aabbccddeeff -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 00112233445566778889aabbccddeeff -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 00112233445566778889aabbccddeeff -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 stream cypher 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 00112233445566778889aabbccdd
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:~$
[09/19/19]seed@VM:~$
[09/19/19]seed@VM:~$
[09/19/19]seed@VM:~$ ls -l -rt 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:~$
```

Of c:

Uncorr upt ed 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
00000002 4F 1D C6 C6 02 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
00000008 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
00000004 87 BF 7A FC 21 FB 5B E3 7E 56 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
0000000b 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

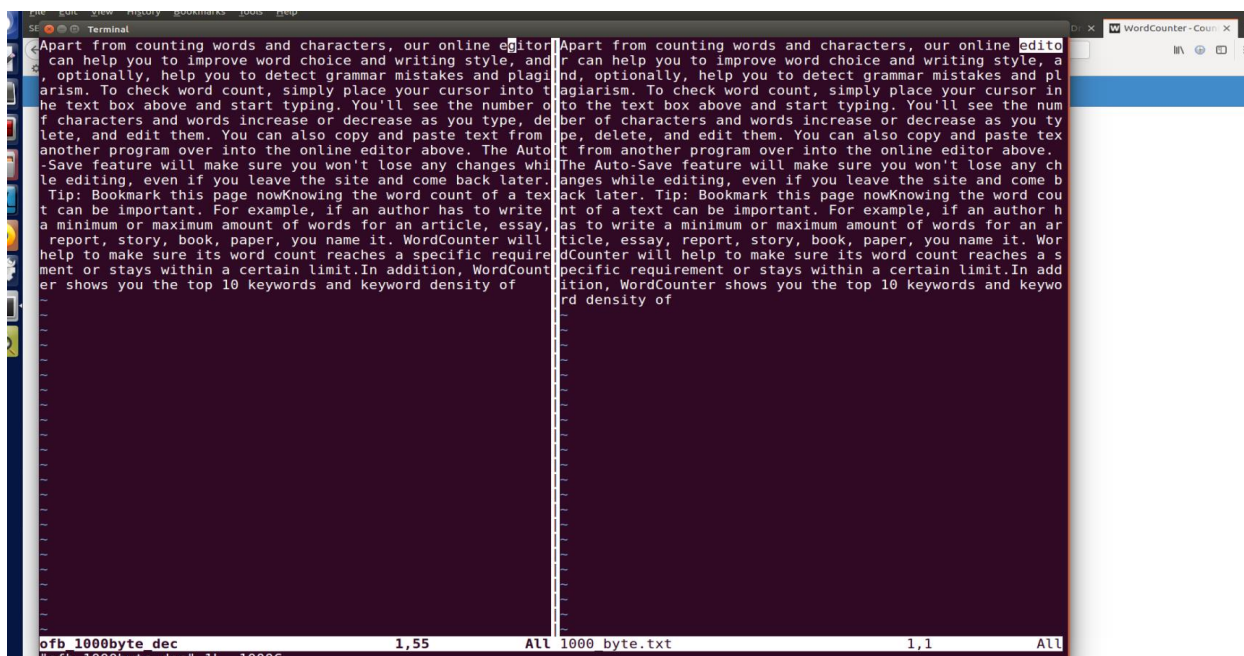
Corrupted one byte at 55th location:

```

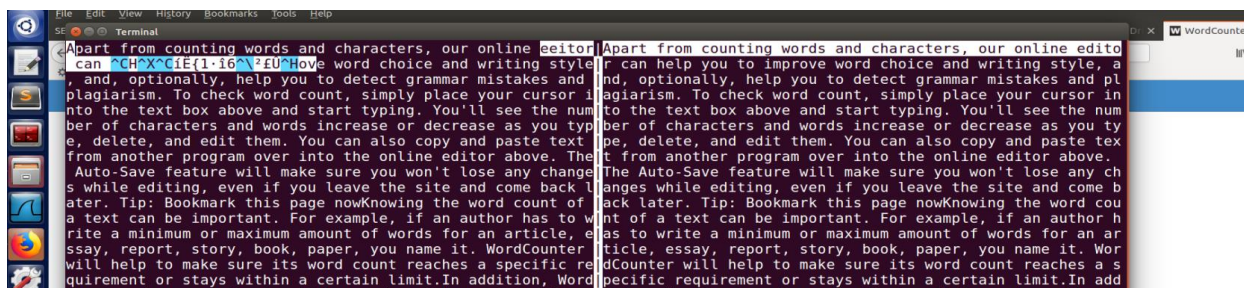
00000000 c6 f6 ee 57 b5 1f dc a3 c0 f3 9f 7e 1a 65 32 19 58 3a fe 27 cc 34 30 97 84 78 67 92 37 49 2b f7 b4 1f 5d 02 11 00 2d 26 39 86 04 f0 ...w.....e2X:'.40_xg.7i+[...
00000002 4f 1d c6 62 6c 42 dc ff a5 f3 8f 4f 1e 13 3a bc 1f 7b 74 15 71 b1 a0 f8 1e 2a 8d 46 2d c3 da ac bb 39 73 2e 05 df bd 03 01 3a ...b.L.E_00:'.t.....'.F.(...
00000008 09 9f de 09 02 7f 50 98 94 31 a6 f1 9e 0a 69 1e 6a 8c 8f a0 e2 9e 4c 7c 54 74 bb 3c d5 13 9d 91 38 6e b3 d6 54 46 de 6a 48 8b ae ...P.l.....t.....t.....8o
0000000a 87 bf 7a fc 21 fb 5b 83 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.l.....k43.....8.....d.m
0000000c c8 f7 f1 c0 48 30 9b 55 c2 2f 05 fc 08 e3 2d fa 6e 72 fa f4 14 2c b5 32 51 57 19 6c 9a 2e 98 47 30 a1 d1 3e 34 1b a6 c8 3d f7 4d b9 ...v/.....nr.....2QW.l.....GZ
0000000e 1a ea ba c1 48 e8 eb ad e2 fc ec 40 fa 2a a2 69 5a 66 8a 80 8e 7f 94 32 75 8f e1 32 10 74 5a b9 85 c1 9c f8 33 af 1a 0a b8 ...R.....8.z.b.fj>.....2u.....t0
00000010 ea 7a eb d8 5f 43 79 67 ec c9 08 70 30 20 5f 4e 55 f8 30 27 67 4d f1 fe ff f7 e3 00 9a 86 7e 04 29 d4 72 76 35 28 8b dd 1c cb 49 f2 ...z.....Cyg.p0_NU0'gM.....).
00000012 38 98 28 42 53 47 87 06 b1 c0 80 31 19 c2 52 65 c6 77 48 c2 fd 0a 1d 3b 3a 8c 08 fe 2a 5f 13 dd 0b 2d 83 a2 43 3a 49 58 ...C.....

```

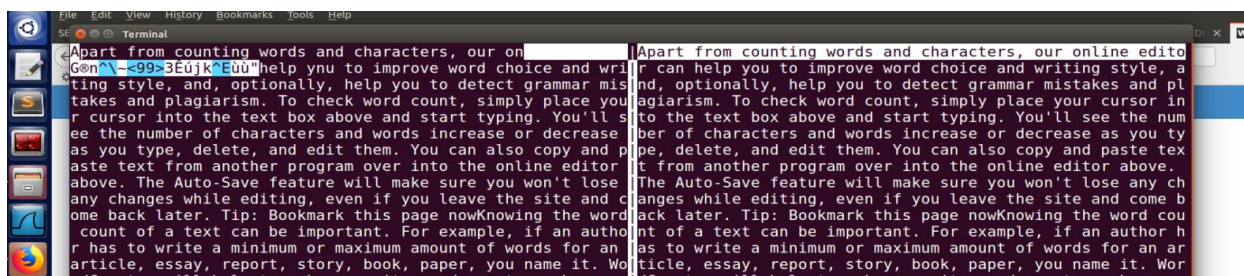

Decrypted and compared: “d” in original became “g”; Only one bit is corrupted in plain text.



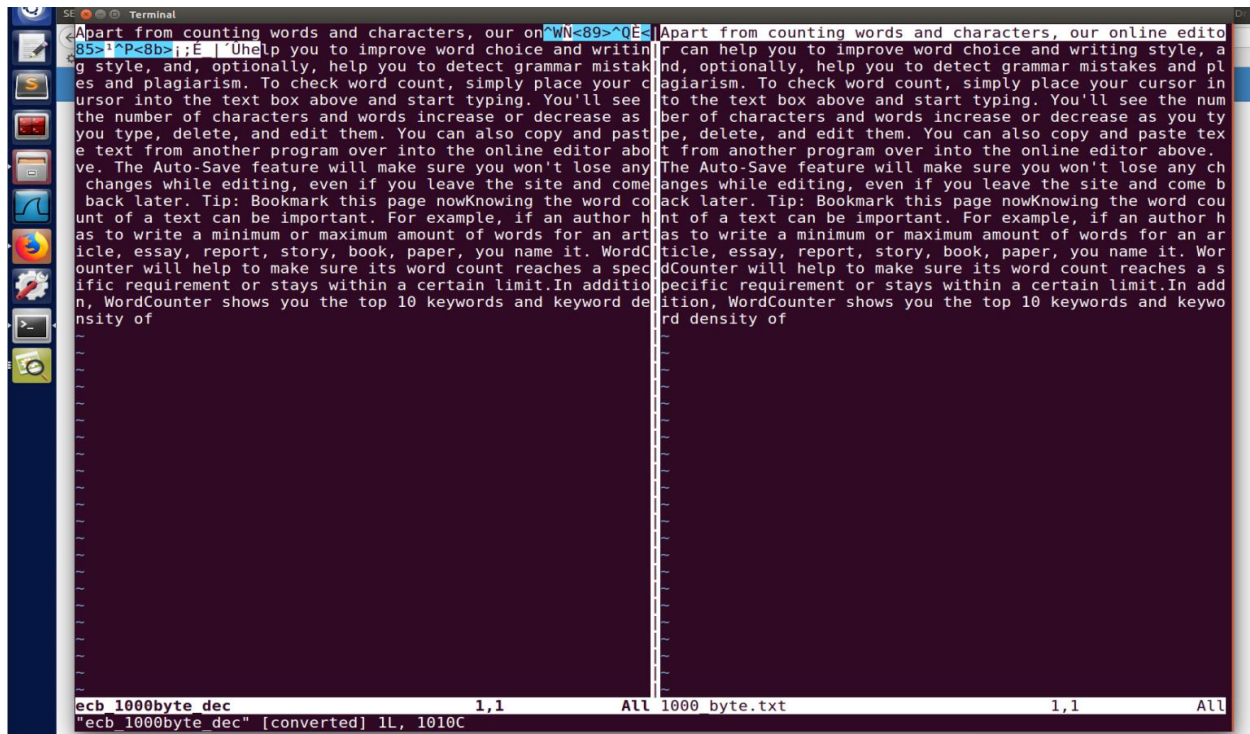
CFB: on changing 1 byte. Whole block got corrupted



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[] = "062ff0112cb32d04d0adcfa02d215abd40a5f932dalebbd3744de5d16be5a4d7";
    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) {
            strcat(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_ptr +=sprintf( enc_ptr,"%02x",outbuf[i]);
}
*(enc_ptr + 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
```

Runni ng

```
[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"

A screenshot of a terminal window with a dark purple background and a blue sidebar on the left containing various application icons. The terminal displays a word search game. It lists words and their corresponding hexadecimal hashes. The words are: sec, secant, secede, secession, seclude, seclusion, second, secondary, secondhand, secrecy, and secret. At the bottom, it shows a long hexadecimal string '062ff0112b32d04d0adcfa02d215abd40a5f932da1ebbd3744de5d16be5a4d7' and the text 'Found the match'. The bottom right corner of the terminal shows '80345,1' and 'Bot'.