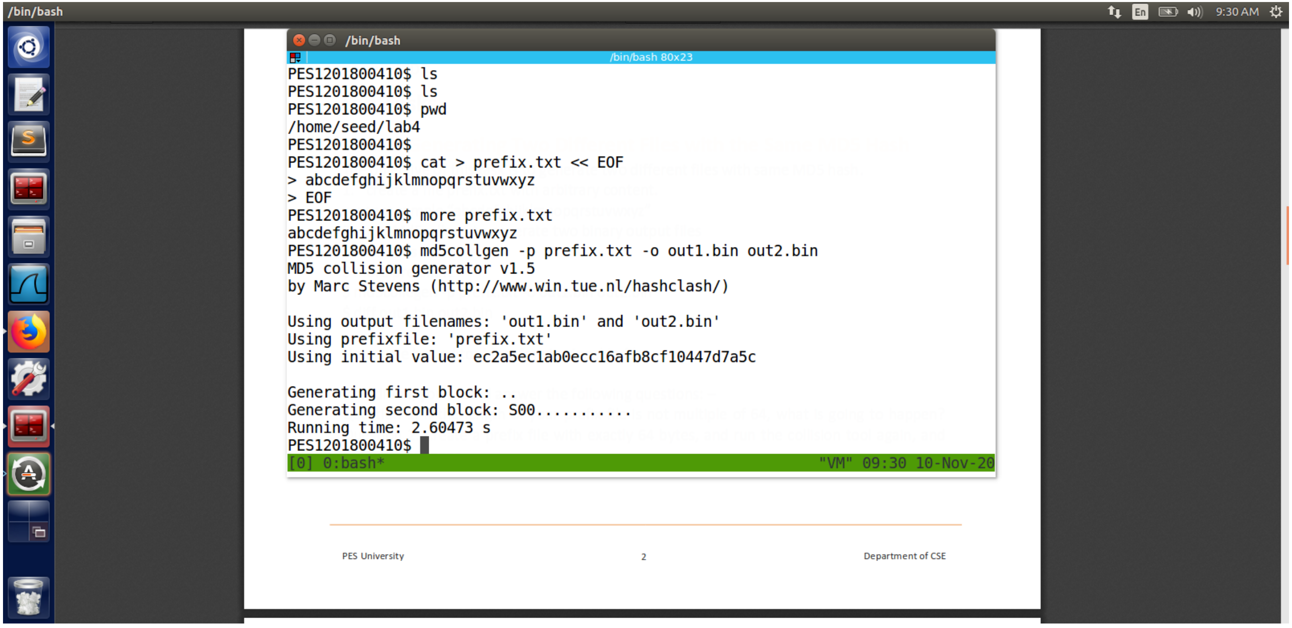
PES1201800410 sem V A R PRASHANTH

lab 4

MD5 Collision Attack Lab

Task 1

generating 2 files with with same md5 hash using md5collegen

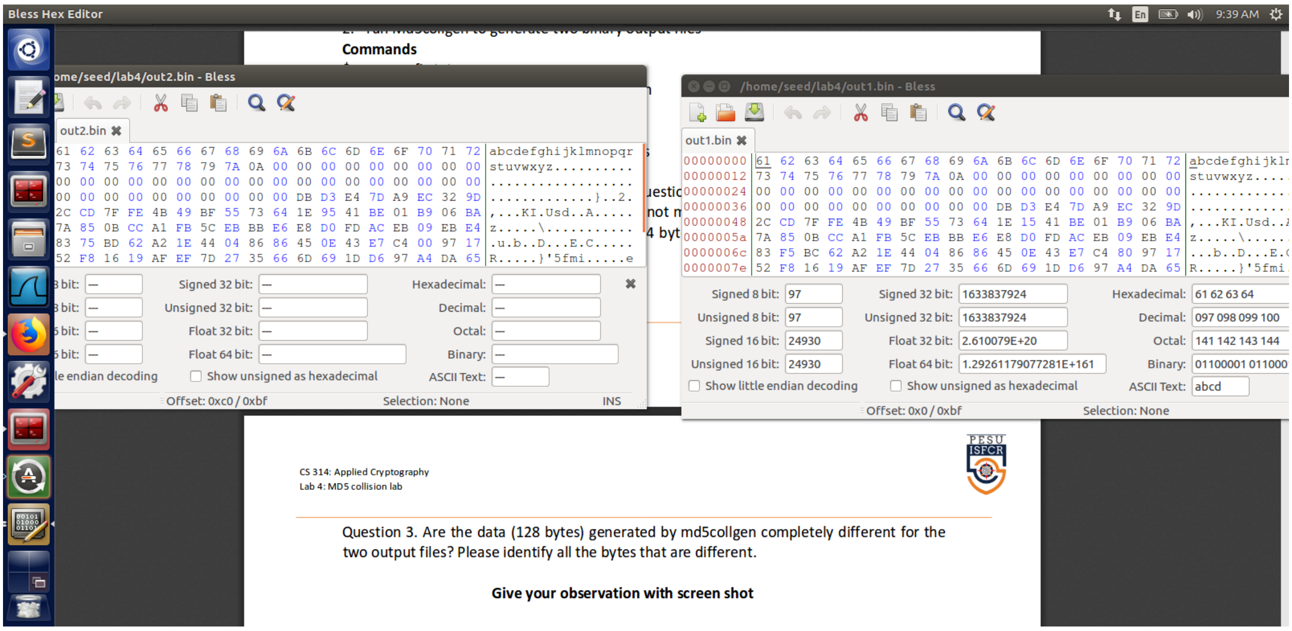


```
PES1201800410$ ls
PES1201800410$ ls
PES1201800410$ pwd
/home/seed/lab4
PES1201800410$
PES1201800410$ cat > prefix.txt << EOF
> abcdefghijklmnopqrstuvwxyz
> EOF
PES1201800410$ more prefix.txt
abcdefghijklmnopqrstuvwxyz
PES1201800410$ md5collgen -p prefix.txt -o out1.bin out2.bin
MD5 collision generator v1.5
by Marc Stevens (http://www.win.tue.nl/hashclash/)

Using output filenames: 'out1.bin' and 'out2.bin'
Using prefixfile: 'prefix.txt'
Using initial value: ec2a5ec1ab0ecc16afb8cf10447d7a5c

Generating first block: ..
Generating second block: S00...........
Running time: 2.60473 s
PES1201800410$
[0] 0:bash*                                          "VM" 09:30 10-Nov-20
```

watch the difference in bless



/home/seed/lab4/out2.bin - Bless

```
61 62 63 64 65 66 67 68 69 6A 6B 6C 6D 6E 6F 70 71 72 abcdefghijklmnopqr
73 74 75 76 77 78 79 7A 0A 00 00 00 00 00 00 00 00 00 stuvwxyz..........
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ..................
00 00 00 00 00 00 00 00 00 00 DB D3 E4 7D A9 EC 32 9D ..............}..2.
2C CD 7F FE 4B 49 BF 55 73 64 1E 95 41 BE 01 B9 06 BA ,...KI.Usd..A.....
7A 85 0B CC A1 FB 5C EB BB E6 E8 D0 FD AC EB 09 EB E4 z.....\...........
83 75 BD 62 A2 1E 44 04 86 86 45 0E 43 E7 C4 00 97 17 .u.b..D...E.C.....
52 F8 16 19 AF EF 7D 27 35 66 6D 69 1D D6 97 A4 DA 65 R.....}'5fmi.....e
```

Offset: 0xc0 / 0xbf Selection: None INS

/home/seed/lab4/out1.bin - Bless

```
00000000 61 62 63 64 65 66 67 68 69 6A 6B 6C 6D 6E 6F 70 71 72 abcdefghijklr
00000012 73 74 75 76 77 78 79 7A 0A 00 00 00 00 00 00 00 00 00 stuvwxyz....
00000024 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ............
00000036 00 00 00 00 00 00 00 00 00 00 DB D3 E4 7D A9 EC 32 9D ............
00000048 2C CD 7F FE 4B 49 BF 55 73 64 1E 15 41 BE 01 B9 06 BA ,...KI.Usd..
0000005a 7A 85 0B CC A1 FB 5C EB BB E6 E8 D0 FD AC EB 09 EB E4 z.....\.....
0000006c 83 F5 BC 62 A2 1E 44 04 86 86 45 0E 43 E7 C4 80 97 17 ...b..D...E.
0000007e 52 F8 16 19 AF EF 7D 27 35 66 6D 69 1D D6 97 A4 DA 65 R.....}'5fmi
```

Signed 8 bit: 97 Signed 32 bit: 1633837924 Hexadecimal: 61 62 63 64

Unsigned 8 bit: 97 Unsigned 32 bit: 1633837924 Decimal: 097 098 099 100

Signed 16 bit: 24930 Float 32 bit: 2.610079E+20 Octal: 141 142 143 144

Unsigned 16 bit: 24930 Float 64 bit: 1.29261179077281E+161 Binary: 01100001 011000

□ Show little endian decoding □ Show unsigned as hexadecimal ASCII Text: abcd

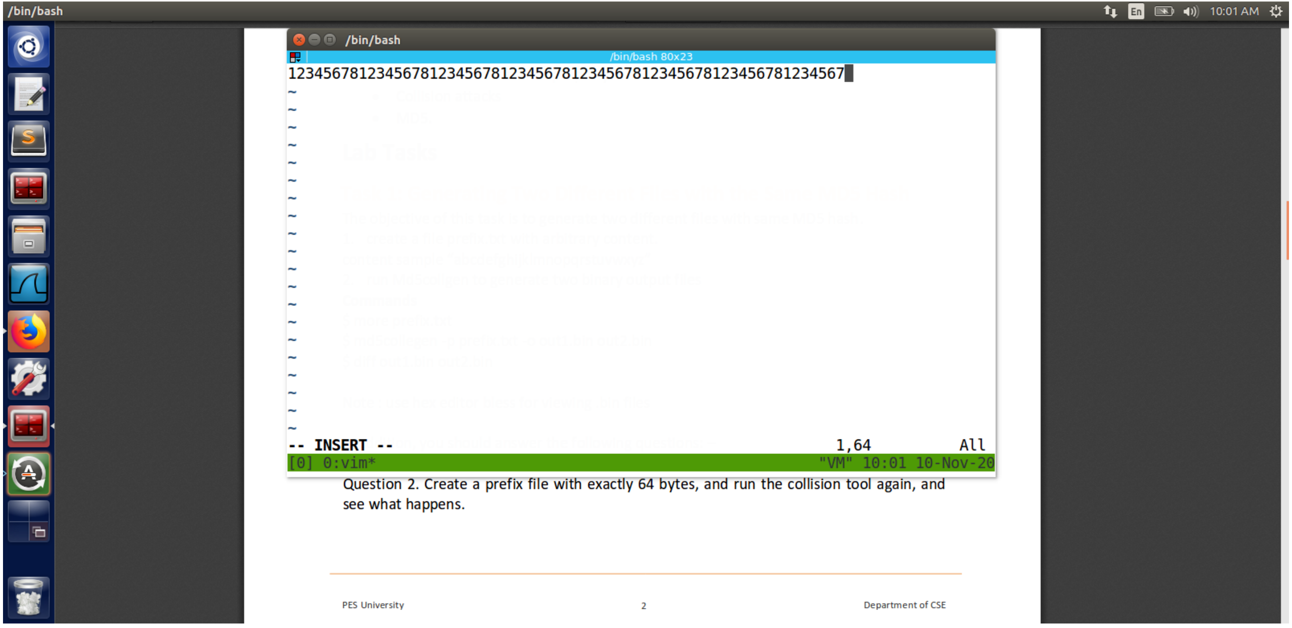
Offset: 0x0 / 0xbf Selection: None

CS 314: Applied Cryptography
Lab 4: MD5 collision lab

Question 3. Are the data (128 bytes) generated by md5collgen completely different for the two output files? Please identify all the bytes that are different.

Give your observation with screen shot

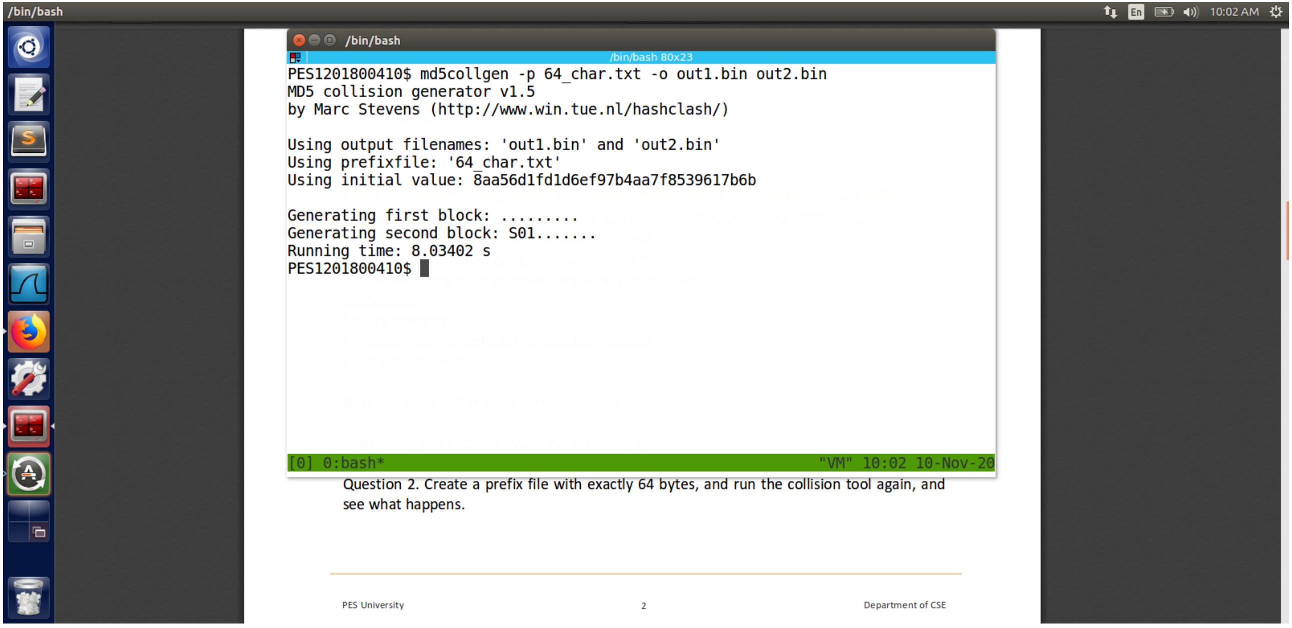
try creating new file with 64 bytes



```
1234567812345678123456781234567812345678123456781234567812345678
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
-- INSERT --                                                  1,64          All
[0] 0:vim*                                               "VM" 10:01 10-Nov-20
```

Question 2. Create a prefix file with exactly 64 bytes, and run the collision tool again, and see what happens.

create 2 files with md5collegen
check difference using bless

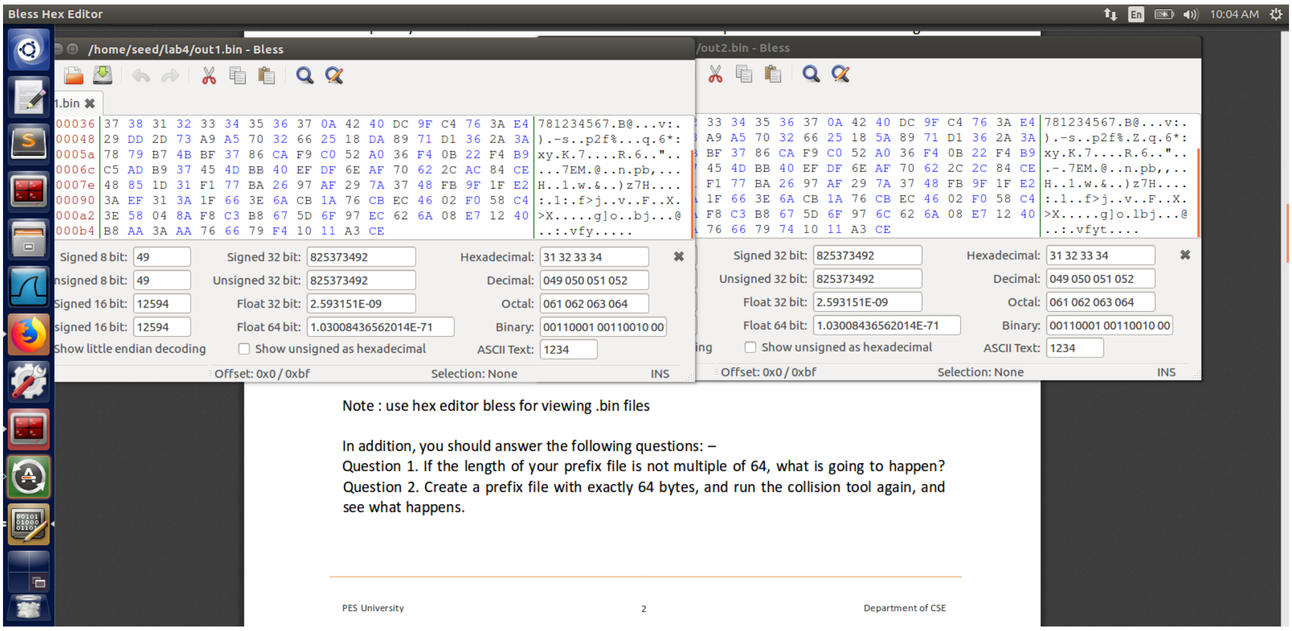


```
PES1201800410$ md5collgen -p 64_char.txt -o out1.bin out2.bin
MD5 collision generator v1.5
by Marc Stevens (http://www.win.tue.nl/hashclash/)

Using output filenames: 'out1.bin' and 'out2.bin'
Using prefixfile: '64_char.txt'
Using initial value: 8aa56d1fd1d6ef97b4aa7f8539617b6b

Generating first block: .........
Generating second block: S01.......
Running time: 8.03402 s
PES1201800410$ 
[0] 0:bash*                                              "VM" 10:02 10-Nov-20
```

Question 2. Create a prefix file with exactly 64 bytes, and run the collision tool again, and see what happens.



Bless Hex Editor

/home/seed/lab4/out1.bin - Bless

```
00036 37 38 31 32 33 34 35 36 37 0A 42 40 DC 9F C4 76 3A E4 781234567.B@...v:.
00048 29 DD 2D 73 A9 A5 70 32 66 25 18 DA 89 71 D1 36 2A 3A ).-s..p2f%...q.6*:
0005a 78 79 B7 4B BF 37 86 CA F9 C0 52 A0 36 F4 0B 22 F4 B9 xy.K.7....R.6.."..
0006c C5 AD B9 37 45 4D BB 40 EF DF 6E AF 70 62 2C AC 84 CE ...7EM.@..n.pb,...
0007e 48 85 1D 31 F1 77 BA 26 97 AF 29 7A 37 48 FB 9F 1F E2 H..1.w.&..)z7H....
00090 3A EF 31 3A 1F 66 3E 6A CB 1A 76 CB EC 46 02 F0 58 C4 :.1:.f>j..v..F..X.
000a2 3E 58 04 8A F8 C3 B8 67 5D 6F 97 EC 62 6A 08 E7 12 40 >X.....g]o..bj...@
000b4 B8 AA 3A AA 76 66 79 F4 10 11 A3 CE                   ..:.vfy.....
```

Signed 8 bit: 49 | Signed 32 bit: 825373492 | Hexadecimal: 31 32 33 34
Unsigned 8 bit: 49 | Unsigned 32 bit: 825373492 | Decimal: 049 050 051 052
Signed 16 bit: 12594 | Float 32 bit: 2.593151E-09 | Octal: 061 062 063 064
Unsigned 16 bit: 12594 | Float 64 bit: 1.03008436562014E-71 | Binary: 00110001 00110010 00
Show little endian decoding | □ Show unsigned as hexadecimal | ASCII Text: 1234

Offset: 0x0 / 0xbf   Selection: None   INS

/out2.bin - Bless

```
33 34 35 36 37 0A 42 40 DC 9F C4 76 3A E4 781234567.B@...v:.
A9 A5 70 32 66 25 18 5A 89 71 D1 36 2A 3A ).-s..p2f%.Z.q.6*:
BF 37 86 CA F9 C0 52 A0 36 F4 0B 22 F4 B9 xy.K.7....R.6.."..
45 4D BB 40 EF DF 6E AF 70 62 2C 2C 84 CE .-.7EM.@..n.pb,,..
F1 77 BA 26 97 AF 29 7A 37 48 FB 9F 1F E2 H..1.w.&..)z7H....
1F 66 3E 6A CB 1A 76 CB EC 46 02 F0 58 C4 :.1..f>j..v..F..X.
F8 C3 B8 67 5D 6F 97 6C 62 6A 08 E7 12 40 >X.....g]o.lbj...@
76 66 79 74 10 11 A3 CE                   ..:.vfyt....
```

Signed 32 bit: 825373492 | Hexadecimal: 31 32 33 34
Unsigned 32 bit: 825373492 | Decimal: 049 050 051 052
Float 32 bit: 2.593151E-09 | Octal: 061 062 063 064
Float 64 bit: 1.03008436562014E-71 | Binary: 00110001 00110010 00
□ Show unsigned as hexadecimal | ASCII Text: 1234

Offset: 0x0 / 0xbf   Selection: None   INS

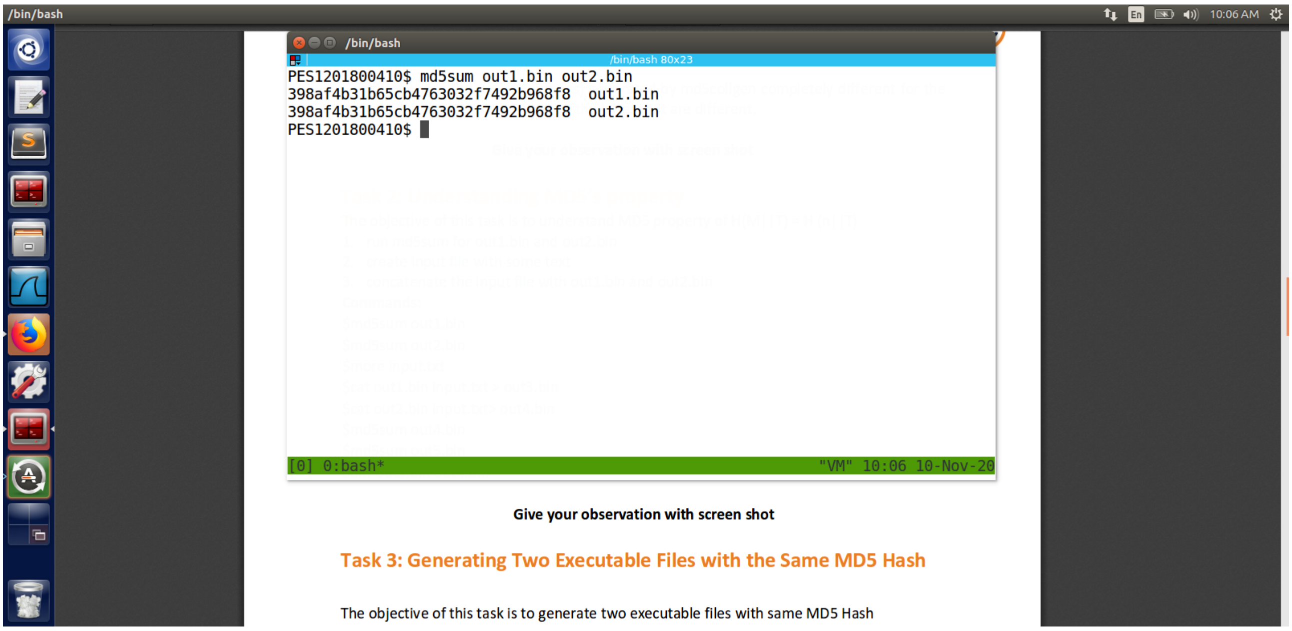
Note : use hex editor bless for viewing .bin files

In addition, you should answer the following questions: –
Question 1. If the length of your prefix file is not multiple of 64, what is going to happen?
Question 2. Create a prefix file with exactly 64 bytes, and run the collision tool again, and see what happens.

PES University 2 Department of CSE

```
we can obeserve that there is no padding of \x00 in the end of the data.
Ans 1: there will be a 00 till the file becomes multiple of 64
And 2: there will be no 00 or padding as it will be multiple of 64
And 3: as we can see there are minor changes
```

```
Task 2
find md5sum of both files
```



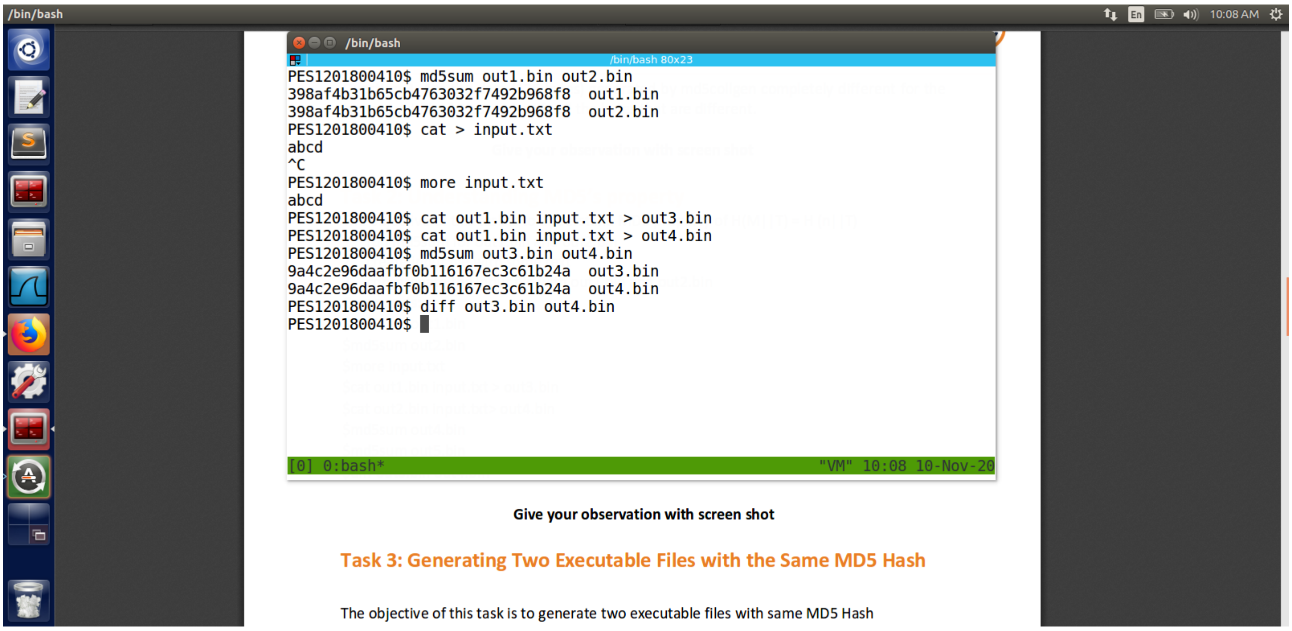
/bin/bash

```
PES1201800410$ md5sum out1.bin out2.bin
398af4b31b65cb4763032f7492b968f8  out1.bin
398af4b31b65cb4763032f7492b968f8  out2.bin
PES1201800410$ 
[0] 0:bash*                                    "VM" 10:06 10-Nov-20
```

Give your observation with screen shot

Task 3: Generating Two Executable Files with the Same MD5 Hash

The objective of this task is to generate two executable files with same MD5 Hash create a file with adding some input string to both the file and check for md5sum



/bin/bash

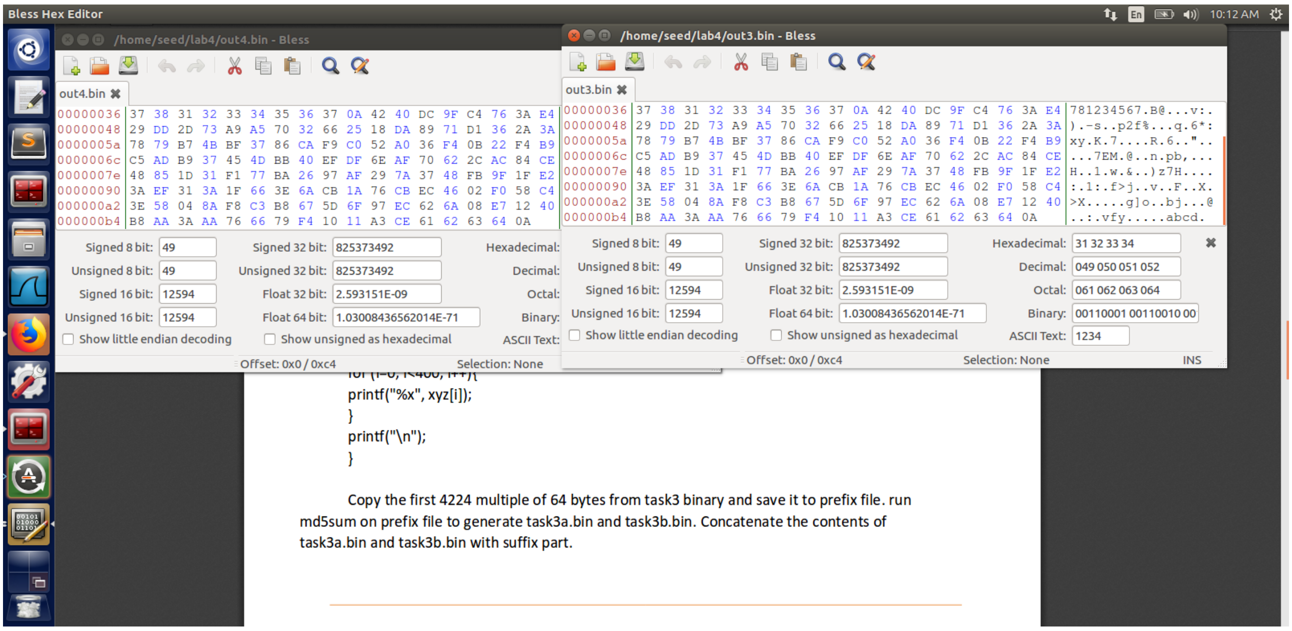
```
PES1201800410$ md5sum out1.bin out2.bin
398af4b31b65cb4763032f7492b968f8  out1.bin
398af4b31b65cb4763032f7492b968f8  out2.bin
PES1201800410$ cat > input.txt
abcd
^C
PES1201800410$ more input.txt
abcd
PES1201800410$ cat out1.bin input.txt > out3.bin
PES1201800410$ cat out1.bin input.txt > out4.bin
PES1201800410$ md5sum out3.bin out4.bin
9a4c2e96daafbf0b116167ec3c61b24a  out3.bin
9a4c2e96daafbf0b116167ec3c61b24a  out4.bin
PES1201800410$ diff out3.bin out4.bin
PES1201800410$ 
```

```
[0] 0:bash*                                          "VM" 10:08 10-Nov-20
```

Give your observation with screen shot

Task 3: Generating Two Executable Files with the Same MD5 Hash

The objective of this task is to generate two executable files with same MD5 Hash



/home/seed/lab4/out4.bin - Bless

out4.bin

```
00000036 37 38 31 32 33 34 35 36 37 0A 42 40 DC 9F C4 76 3A E4
00000048 29 DD 2D 73 A9 A5 70 32 66 25 18 DA 89 71 D1 36 2A 3A
0000005a 78 79 B7 4B BF 37 86 CA F9 C0 52 A0 36 F4 0B 22 F4 B9
0000006c C5 AD B9 37 45 4D BB 40 EF DF 6E AF 70 62 2C AC 84 CE
0000007e 48 85 1D 31 F1 77 BA 26 97 AF 29 7A 37 48 FB 9F 1F E2
00000090 3A EF 31 3A 1F 66 3E 6A CB 1A 76 CB EC 46 02 F0 58 C4
000000a2 3E 58 04 8A F8 C3 B8 67 5D 6F 97 EC 62 6A 08 E7 12 40
000000b4 B8 AA 3A AA 76 66 79 F4 10 11 A3 CE 61 62 63 64 0A
```

Signed 8 bit: 49 Signed 32 bit: 825373492 Hexadecimal:
Unsigned 8 bit: 49 Unsigned 32 bit: 825373492 Decimal:
Signed 16 bit: 12594 Float 32 bit: 2.593151E-09 Octal:
Unsigned 16 bit: 12594 Float 64 bit: 1.03008436562014E-71 Binary:
☐ Show little endian decoding ☐ Show unsigned as hexadecimal ASCII Text:

Offset: 0x0 / 0xc4 Selection: None

/home/seed/lab4/out3.bin - Bless

out3.bin

```
00000036 37 38 31 32 33 34 35 36 37 0A 42 40 DC 9F C4 76 3A E4 781234567.B@...v:.
00000048 29 DD 2D 73 A9 A5 70 32 66 25 18 DA 89 71 D1 36 2A 3A ).-s..p2f%...q.6*:
0000005a 78 79 B7 4B BF 37 86 CA F9 C0 52 A0 36 F4 0B 22 F4 B9 xy.K.7....R.6.."..
0000006c C5 AD B9 37 45 4D BB 40 EF DF 6E AF 70 62 2C AC 84 CE ...7EM.@..n.pb,...
0000007e 48 85 1D 31 F1 77 BA 26 97 AF 29 7A 37 48 FB 9F 1F E2 H..1.w.&..)z7H....
00000090 3A EF 31 3A 1F 66 3E 6A CB 1A 76 CB EC 46 02 F0 58 C4 :.1:.f>j..v..F..X.
000000a2 3E 58 04 8A F8 C3 B8 67 5D 6F 97 EC 62 6A 08 E7 12 40 >X.....g]o..bj...@
000000b4 B8 AA 3A AA 76 66 79 F4 10 11 A3 CE 61 62 63 64 0A    ..:.vfy.....abcd.
```

Signed 8 bit: 49 Signed 32 bit: 825373492 Hexadecimal: 31 32 33 34
Unsigned 8 bit: 49 Unsigned 32 bit: 825373492 Decimal: 049 050 051 052
Signed 16 bit: 12594 Float 32 bit: 2.593151E-09 Octal: 061 062 063 064
Unsigned 16 bit: 12594 Float 64 bit: 1.03008436562014E-71 Binary: 00110001 00110010 00
☐ Show little endian decoding ☐ Show unsigned as hexadecimal ASCII Text: 1234

Offset: 0x0 / 0xc4 Selection: None INS

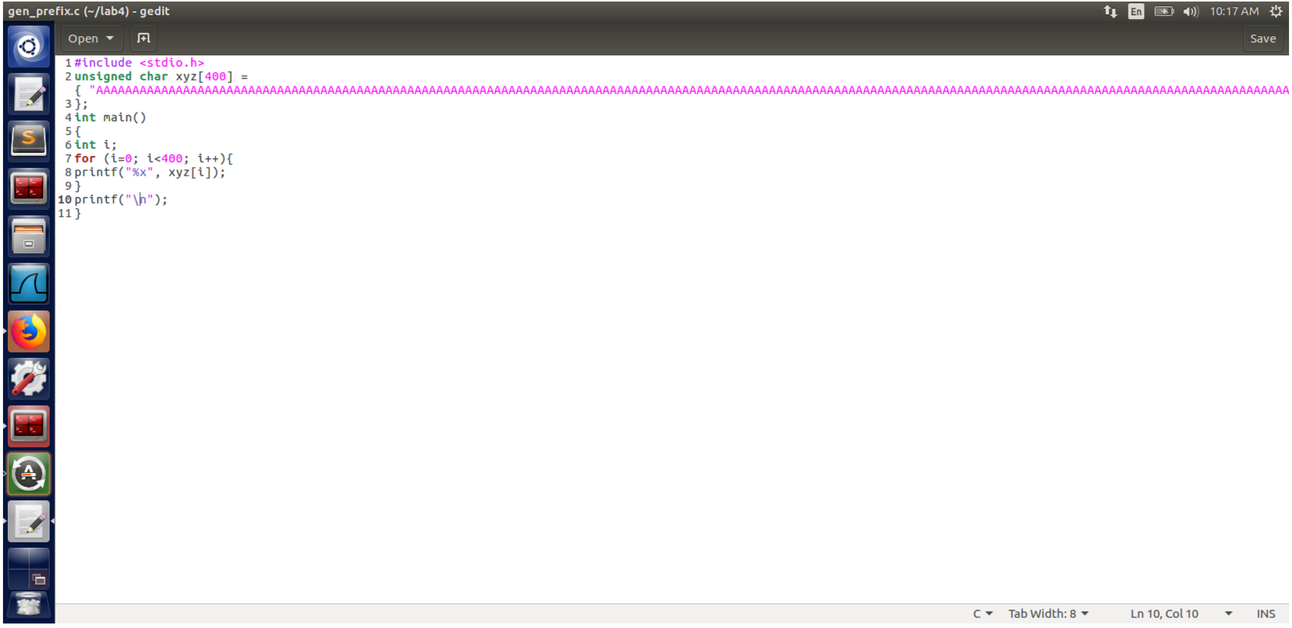
```
for (i=0; i<400; i++){
printf("%x", xyz[i]);
}
printf("\n");
}
```

Copy the first 4224 multiple of 64 bytes from task3 binary and save it to prefix file. run md5sum on prefix file to generate task3a.bin and task3b.bin. Concatenate the contents of task3a.bin and task3b.bin with suffix part.

Task 3

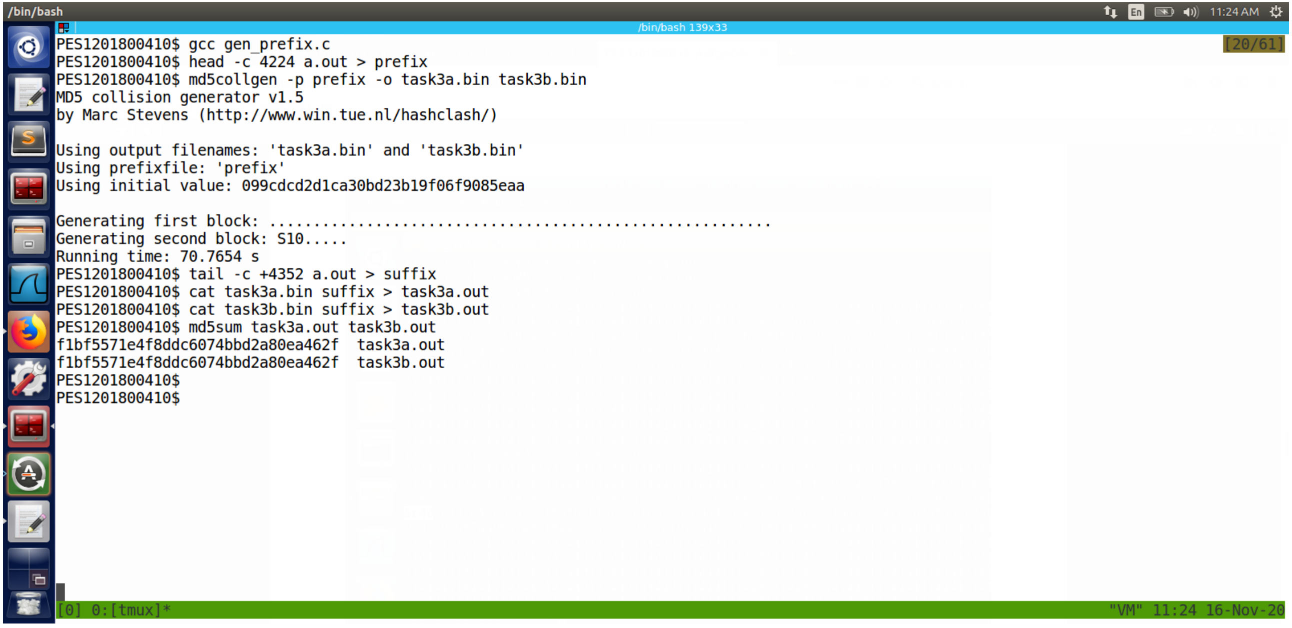
compile gen_prefix.c file given in the manual .

gen_prefix source code



```c
#include <stdio.h>
unsigned char xyz[400] =
 { "AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
};
int main()
{
int i;
for (i=0; i<400; i++){
printf("%x", xyz[i]);
}
printf("\n");
}
```

Cut 4224 bites and create a md5collegen

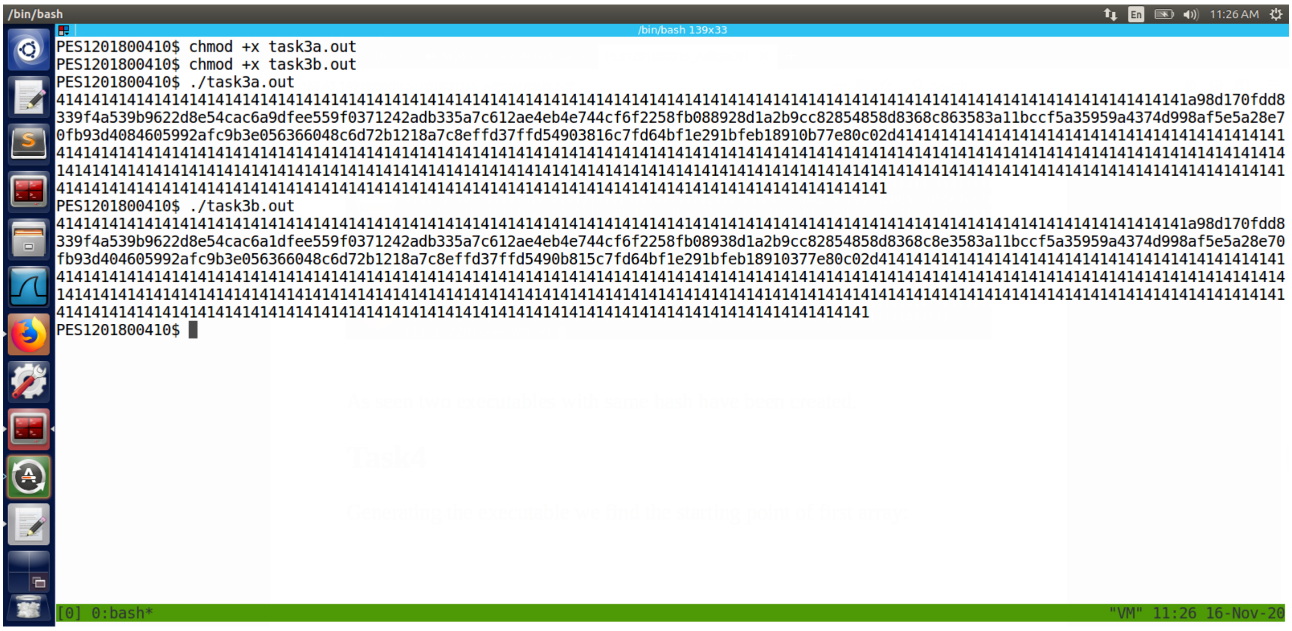


```
PES1201800410$ gcc gen_prefix.c
PES1201800410$ head -c 4224 a.out > prefix
PES1201800410$ md5collgen -p prefix -o task3a.bin task3b.bin
MD5 collision generator v1.5
by Marc Stevens (http://www.win.tue.nl/hashclash/)

Using output filenames: 'task3a.bin' and 'task3b.bin'
Using prefixfile: 'prefix'
Using initial value: 099cdcd2d1ca30bd23b19f06f9085eaa

Generating first block: ......................................................
Generating second block: S10.....
Running time: 70.7654 s
PES1201800410$ tail -c +4352 a.out > suffix
PES1201800410$ cat task3a.bin suffix > task3a.out
PES1201800410$ cat task3b.bin suffix > task3b.out
PES1201800410$ md5sum task3a.out task3b.out
f1bf5571e4f8ddc6074bbd2a80ea462f  task3a.out
f1bf5571e4f8ddc6074bbd2a80ea462f  task3b.out
PES1201800410$
PES1201800410$
```

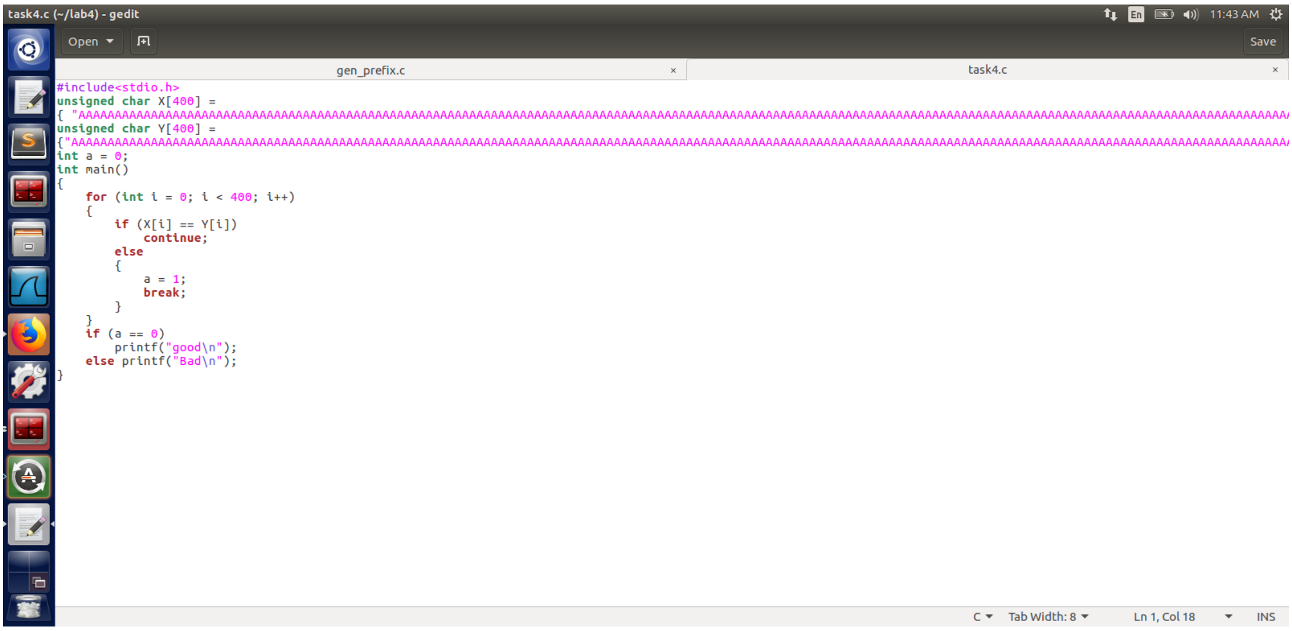
execute and check for differences in both the programms



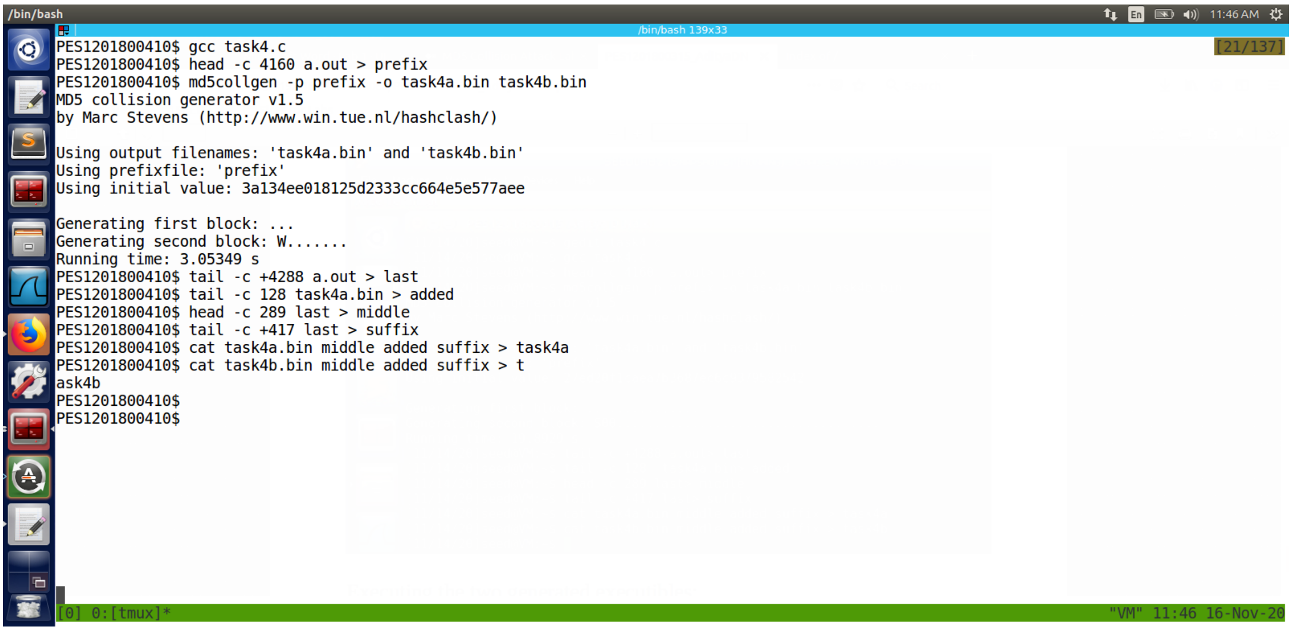
```
PES1201800410$ chmod +x task3a.out
PES1201800410$ chmod +x task3b.out
PES1201800410$ ./task3a.out
41414141414141414141414141414141414141414141414141414141414141414141414141414141414141414141414141414141414141414141414141414141a98d170fdd8
339f4a539b9622d8e54cac6a9dfee559f0371242adb335a7c612ae4eb4e744cf6f2258fb088928d1a2b9cc82854858d8368c863583a11bccf5a35959a4374d998af5e5a28e7
0fb93d4084605992afc9b3e056366048c6d72b1218a7c8effd37ffd54903816c7fd64bf1e291bfeb18910b77e80c02d41414141414141414141414141414141414141414141
4141414141414141414141414141414141414141414141414141414141414141414141414141414141414141414141414141414141414141414141414141414141414141414
1414141414141414141414141414141414141414141414141414141414141414141414141414141414141414141414141414141414141414141414141414141414141414141
4141414141414141414141414141414141414141414141414141414141414141414141414141414141414141414141
PES1201800410$ ./task3b.out
41414141414141414141414141414141414141414141414141414141414141414141414141414141414141414141414141414141414141414141414141414141a98d170fdd8
339f4a539b9622d8e54cac6a1dfee559f0371242adb335a7c612ae4eb4e744cf6f2258fb08938d1a2b9cc82854858d8368c8e3583a11bccf5a35959a4374d998af5e5a28e70
fb93d404605992afc9b3e056366048c6d72b1218a7c8effd37ffd5490b815c7fd64bf1e291bfeb18910377e80c02d4141414141414141414141414141414141414141414141
4141414141414141414141414141414141414141414141414141414141414141414141414141414141414141414141414141414141414141414141414141414141414141414
1414141414141414141414141414141414141414141414141414141414141414141414141414141414141414141414141414141414141414141414141414141414141414141
41414141414141414141414141414141414141414141414141414141414141414141414141414141414141414141
PES1201800410$ 
[0] 0:bash*                                                                                                          "VM" 11:26 16-Nov-20
```

Task 4

compile the source code given
task4.c



```
#include<stdio.h>
unsigned char X[400] =
{ "AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
unsigned char Y[400] =
{"AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
int a = 0;
int main()
{
    for (int i = 0; i < 400; i++)
    {
        if (X[i] == Y[i])
            continue;
        else
        {
            a = 1;
            break;
        }
    }
    if (a == 0)
        printf("good\n");
    else printf("Bad\n");
}
```

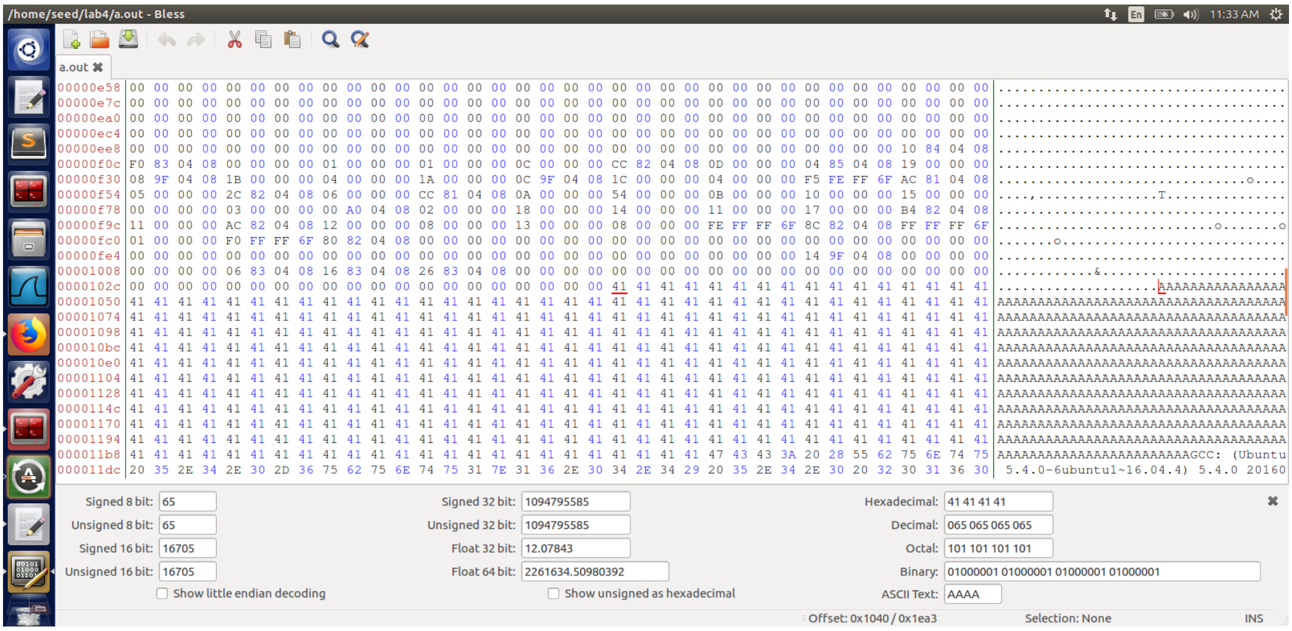


```
PES1201800410$ gcc task4.c
PES1201800410$ head -c 4160 a.out > prefix
PES1201800410$ md5collgen -p prefix -o task4a.bin task4b.bin
MD5 collision generator v1.5
by Marc Stevens (http://www.win.tue.nl/hashclash/)

Using output filenames: 'task4a.bin' and 'task4b.bin'
Using prefixfile: 'prefix'
Using initial value: 3a134ee018125d2333cc664e5e577aee

Generating first block: ...
Generating second block: W.......
Running time: 3.05349 s
PES1201800410$ tail -c +4288 a.out > last
PES1201800410$ tail -c 128 task4a.bin > added
PES1201800410$ head -c 289 last > middle
PES1201800410$ tail -c +417 last > suffix
PES1201800410$ cat task4a.bin middle added suffix > task4a
PES1201800410$ cat task4b.bin middle added suffix > t
ask4b
PES1201800410$
PES1201800410$
[0] 0:[tmux]*                                                "VM" 11:46 16-Nov-20
```

bless of a.out



```
00000e58 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ....................................
00000e7c 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ....................................
00000ea0 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ....................................
00000ec4 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ....................................
00000ee8 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 10 84 04 08 ....................................
00000f0c F0 83 04 08 00 00 00 00 01 00 00 00 01 00 00 00 0C 00 00 00 CC 82 04 08 0D 00 00 00 04 85 04 08 19 00 00 00 ....................................
00000f30 08 9F 04 08 1B 00 00 00 04 00 00 00 1A 00 00 00 0C 9F 04 08 1C 00 00 00 04 00 00 00 F5 FE FF 6F AC 81 04 08 ..............................o.....
00000f54 05 00 00 00 2C 82 04 08 06 00 00 00 CC 81 04 08 0A 00 00 00 54 00 00 00 0B 00 00 00 10 00 00 00 15 00 00 00 ....,...............T...............
00000f78 00 00 00 00 03 00 00 00 00 A0 04 08 02 00 00 00 18 00 00 00 14 00 00 00 11 00 00 00 17 00 00 00 B4 82 04 08 ....................................
00000f9c 11 00 00 00 AC 82 04 08 12 00 00 00 08 00 00 00 13 00 00 00 08 00 00 00 FE FF FF 6F 8C 82 04 08 FF FF FF 6F ...........................o.......o
00000fc0 01 00 00 00 F0 FF FF 6F 80 82 04 08 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 .......o............................
00000fe4 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 14 9F 04 08 00 00 00 00 ....................................
00001008 00 00 00 00 06 83 04 08 16 83 04 08 26 83 04 08 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ............&.......................
0000102c 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 ....................AAAAAAAAAAAAAAAA
00001050 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
00001074 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
00001098 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
000010bc 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
000010e0 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
00001104 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
00001128 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
0000114c 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
00001170 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
00001194 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
000011b8 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 47 43 43 3A 20 28 55 62 75 6E 74 75 AAAAAAAAAAAAAAAAAAAAAAAAGCC: (Ubuntu
000011dc 20 35 2E 34 2E 30 2D 36 75 62 75 6E 74 75 31 7E 31 36 2E 30 34 2E 34 29 20 35 2E 34 2E 30 20 32 30 31 36 30  5.4.0-6ubuntu1~16.04.4) 5.4.0 20160
```

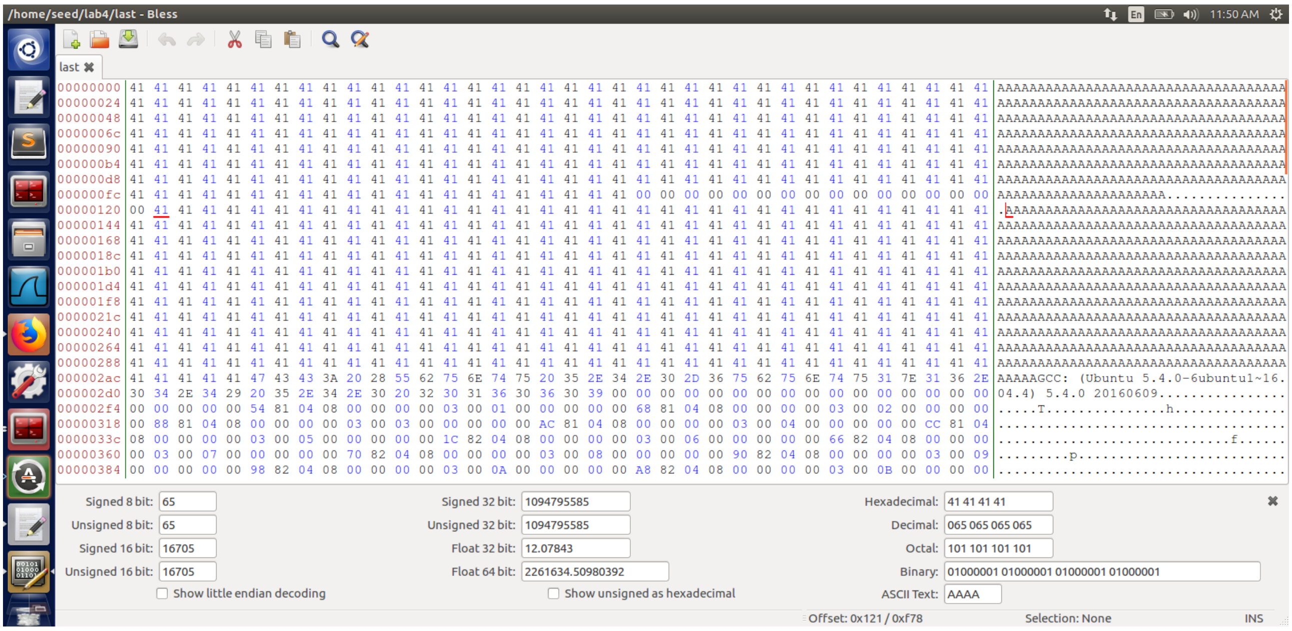
Signed 8 bit: 65
Unsigned 8 bit: 65
Signed 16 bit: 16705
Unsigned 16 bit: 16705
☐ Show little endian decoding

Signed 32 bit: 1094795585
Unsigned 32 bit: 1094795585
Float 32 bit: 12.07843
Float 64 bit: 2261634.50980392
☐ Show unsigned as hexadecimal

Hexadecimal: 41 41 41 41
Decimal: 065 065 065 065
Octal: 101 101 101 101
Binary: 01000001 01000001 01000001 01000001
ASCII Text: AAAA

Offset: 0x1040 / 0x1ea3   Selection: None   INS

bless of last

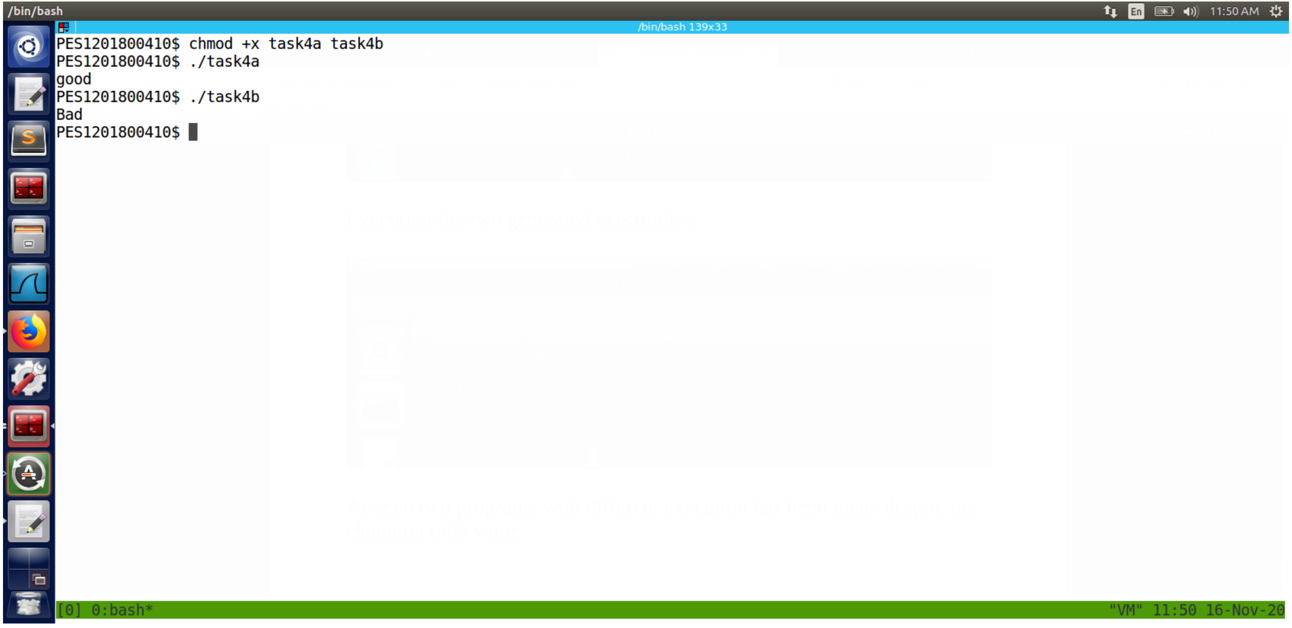


```
00000000 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
00000024 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
00000048 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
0000006c 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
00000090 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
000000b4 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
000000d8 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
000000fc 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 AAAAAAAAAAAAAAAAAAAA................
00000120 00 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 .AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
00000144 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
00000168 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
0000018c 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
000001b0 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
000001d4 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
000001f8 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
0000021c 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
00000240 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
00000264 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
00000288 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
000002ac 41 41 41 41 41 47 43 43 3A 20 28 55 62 75 6E 74 75 20 35 2E 34 2E 30 2D 36 75 62 75 6E 74 75 31 7E 31 36 2E AAAAAGCC: (Ubuntu 5.4.0-6ubuntu1~16.
000002d0 30 34 2E 34 29 20 35 2E 34 2E 30 20 32 30 31 36 30 36 30 39 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 04.4) 5.4.0 20160609................
000002f4 00 00 00 00 00 54 81 04 08 00 00 00 00 03 00 01 00 00 00 00 00 68 81 04 08 00 00 00 00 03 00 02 00 00 00 00 .....T...............h..............
00000318 00 88 81 04 08 00 00 00 00 03 00 03 00 00 00 00 00 AC 81 04 08 00 00 00 00 03 00 04 00 00 00 00 00 CC 81 04 ....................................
0000033c 08 00 00 00 00 03 00 05 00 00 00 00 00 1C 82 04 08 00 00 00 00 03 00 06 00 00 00 00 00 66 82 04 08 00 00 00 .............................f......
00000360 00 03 00 07 00 00 00 00 00 70 82 04 08 00 00 00 00 03 00 08 00 00 00 00 00 90 82 04 08 00 00 00 00 03 00 09 .........p..........................
00000384 00 00 00 00 00 98 82 04 08 00 00 00 00 03 00 0A 00 00 00 00 00 A8 82 04 08 00 00 00 00 03 00 0B 00 00 00 00 ....................................
```

Signed 8 bit:	65	Signed 32 bit:	1094795585	Hexadecimal:	41 41 41 41
Unsigned 8 bit:	65	Unsigned 32 bit:	1094795585	Decimal:	065 065 065 065
Signed 16 bit:	16705	Float 32 bit:	12.07843	Octal:	101 101 101 101
Unsigned 16 bit:	16705	Float 64 bit:	2261634.50980392	Binary:	01000001 01000001 01000001 01000001
	□ Show little endian decoding		□ Show unsigned as hexadecimal	ASCII Text:	AAAA

Offset: 0x121 / 0xf78 Selection: None INS

```
execute and see the differences
if there is any changes in the file it will print bad else good
```



```
PES1201800410$ chmod +x task4a task4b
PES1201800410$ ./task4a
good
PES1201800410$ ./task4b
Bad
PES1201800410$ █
```

```
[0] 0:bash*                                                    "VM" 11:50 16-Nov-20
```