Benjamin Haedt

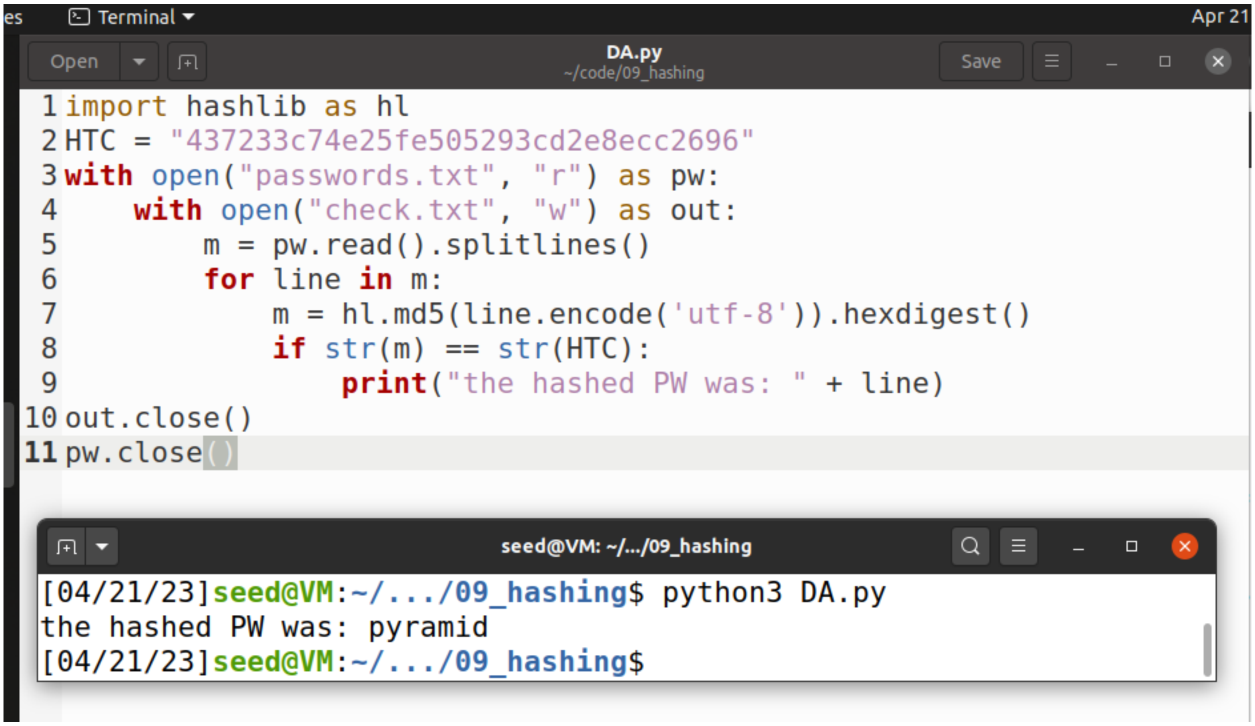
CSCI 476- Computer Security

20 April 2023

Lab – 9 – Hashing

Task 1

Whew, got it, took me to long to program that thing. I just spaced that HTC is already in our hex format. But, there it is, all done.

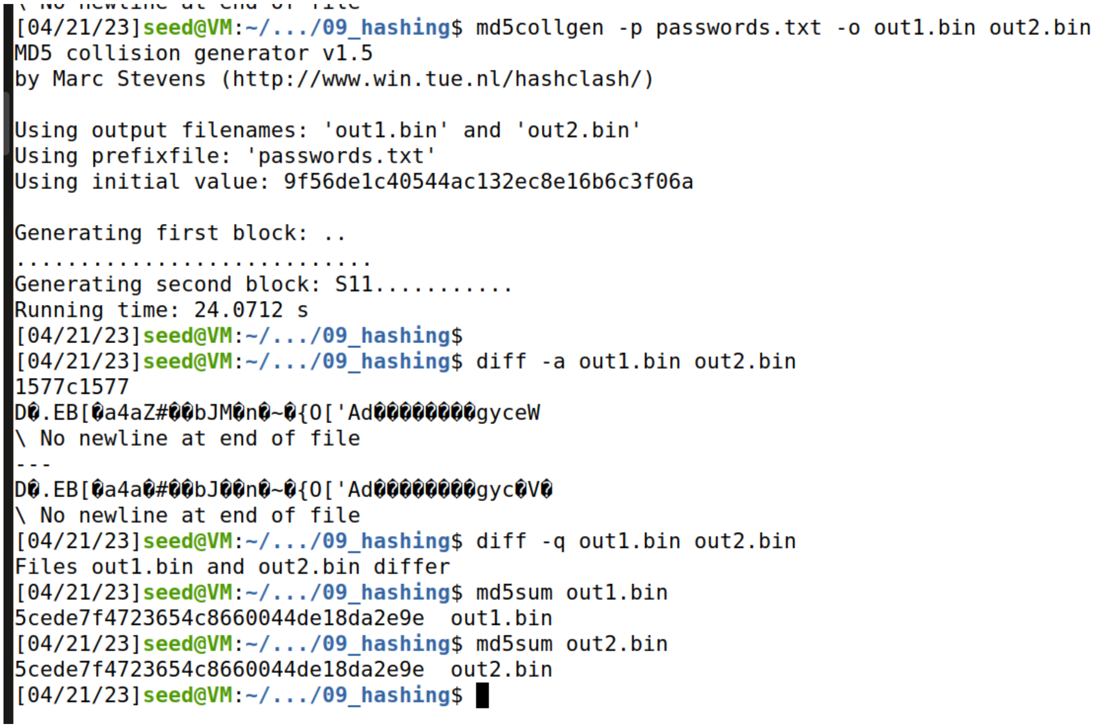


```python
import hashlib as hl
HTC = "437233c74e25fe505293cd2e8ecc2696"
with open("passwords.txt", "r") as pw:
    with open("check.txt", "w") as out:
        m = pw.read().splitlines()
        for line in m:
            m = hl.md5(line.encode('utf-8')).hexdigest()
            if str(m) == str(HTC):
                print("the hashed PW was:  " + line)
out.close()
pw.close()
```

```
[04/21/23]seed@VM:~/.../09_hashing$ python3 DA.py
the hashed PW was: pyramid
[04/21/23]seed@VM:~/.../09_hashing$
```

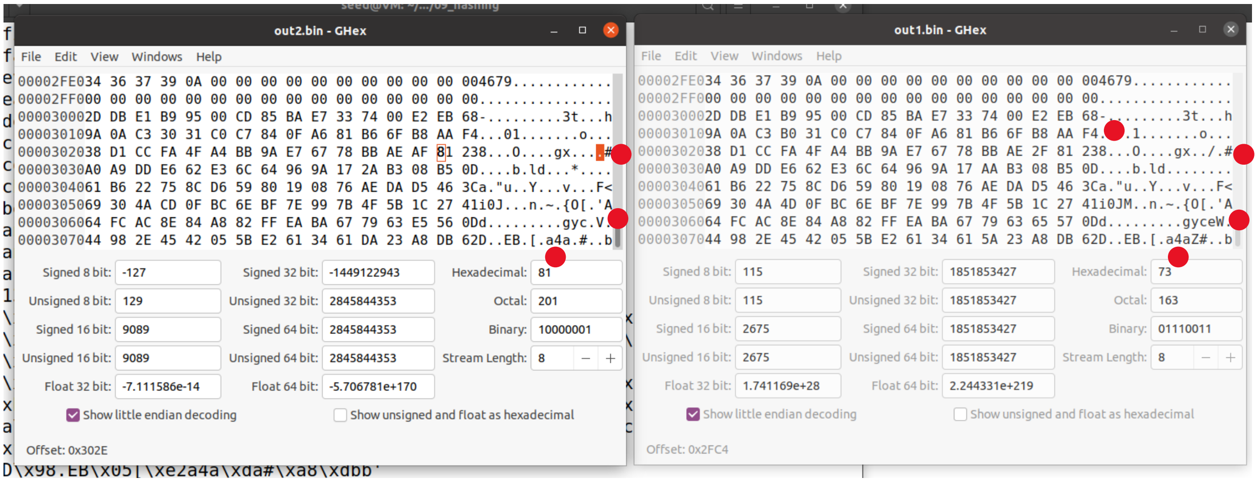
Task 2

When using diff on both out1.bin and out2.bin, we can clearly see that there are differences, but they have the same md5sum.



```
\ No newline at end of file
[04/21/23]seed@VM:~/.../09_hashing$ md5collgen -p passwords.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: 'passwords.txt'
Using initial value: 9f56de1c40544ac132ec8e16b6c3f06a

Generating first block: ..
............................
Generating second block: S11...........
Running time: 24.0712 s
[04/21/23]seed@VM:~/.../09_hashing$
[04/21/23]seed@VM:~/.../09_hashing$ diff -a out1.bin out2.bin
1577c1577
D�.EB[�a4aZ#��bJM�n�~�{O['Ad���������gyceW
\ No newline at end of file
---
D�.EB[�a4a�#��bJ��n�~�{O['Ad���������gyc�V�
\ No newline at end of file
[04/21/23]seed@VM:~/.../09_hashing$ diff -q out1.bin out2.bin
Files out1.bin and out2.bin differ
[04/21/23]seed@VM:~/.../09_hashing$ md5sum out1.bin
5cede7f4723654c8660044de18da2e9e  out1.bin
[04/21/23]seed@VM:~/.../09_hashing$ md5sum out2.bin
5cede7f4723654c8660044de18da2e9e  out2.bin
[04/21/23]seed@VM:~/.../09_hashing$ 
```

Next to the red dots, are some areas that are not the same.



```
00002FE034 36 37 39 0A 00 00 00 00 00 00 00 00 00 00 004679...........
00002FF000 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00................
000030002D DB E1 B9 95 00 CD 85 BA E7 33 74 00 E2 EB 68-.........3t...h
000030109A 0A C3 30 31 C0 C7 84 0F A6 81 B6 6F B8 AA F4...01.......o...
0000302038 D1 CC FA 4F A4 BB 9A E7 67 78 BB AE AF 81 238...O....gx....#
00003030A0 A9 DD E6 62 E3 6C 64 96 9A 17 2A B3 08 B5 0D....b.ld...*....
0000304061 B6 22 75 8C D6 59 80 19 08 76 AE DA D5 46 3Ca."u..Y...v...F<
0000305069 30 4A CD 0F BC 6E BF 7E 99 7B 4F 5B 1C 27 41i0J...n.~.{O[.'A
0000306064 FC AC 8E 84 A8 82 FF EA BA 67 79 63 E5 56 0Dd.........gyc.V.
0000307044 98 2E 45 42 05 5B E2 61 34 61 DA 23 A8 DB 62D..EB.[.a4a.#..b
```

Signed 8 bit: -127 Signed 32 bit: -1449122943 Hexadecimal: 81
Unsigned 8 bit: 129 Unsigned 32 bit: 2845844353 Octal: 201
Signed 16 bit: 9089 Signed 64 bit: 2845844353 Binary: 10000001
Unsigned 16 bit: 9089 Unsigned 64 bit: 2845844353 Stream Length: 8
Float 32 bit: -7.111586e-14 Float 64 bit: -5.706781e+170
☑ Show little endian decoding ☐ Show unsigned and float as hexadecimal
Offset: 0x302E

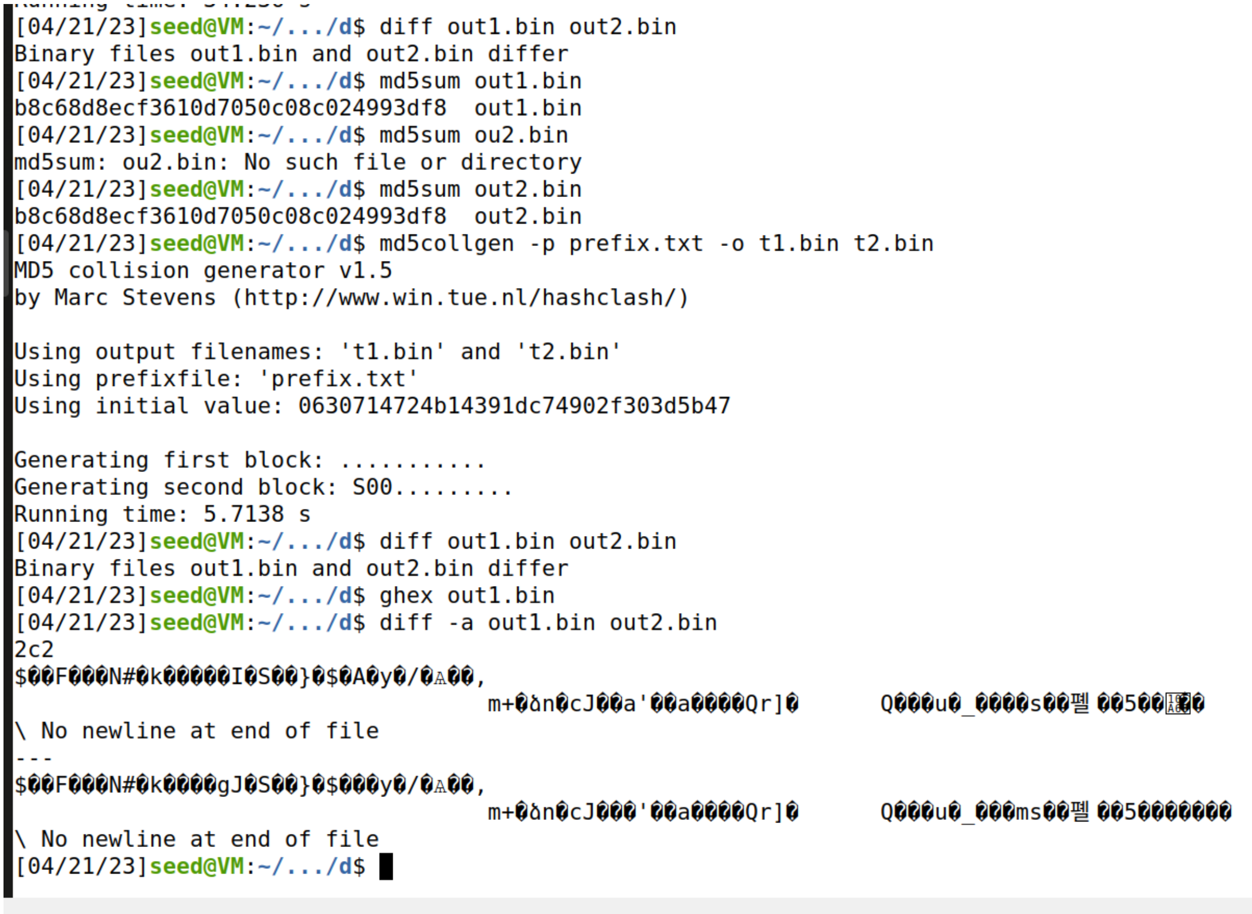
```
00002FE034 36 37 39 0A 00 00 00 00 00 00 00 00 00 00 004679...........
00002FF000 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00................
000030002D DB E1 B9 95 00 CD 85 BA E7 33 74 00 E2 EB 68-.........3t...h
000030109A 0A C3 B0 31 C0 C7 84 0F A6 81 B6 6F B8 AA F4....1.......o...
0000302038 D1 CC FA 4F A4 BB 9A E7 67 78 BB AE 2F 81 238...O....gx../.#
00003030A0 A9 DD E6 62 E3 6C 64 96 9A 17 AA B3 08 B5 0D....b.ld........
0000304061 B6 22 75 8C D6 59 80 19 08 76 AE DA D5 46 3Ca."u..Y...v...F<
0000305069 30 4A 4D 0F BC 6E BF 7E 99 7B 4F 5B 1C 27 41i0JM..n.~.{O[.'A
0000306064 FC AC 8E 84 A8 82 FF EA BA 67 79 63 65 57 0Dd.........gyceW.
0000307044 98 2E 45 42 05 5B E2 61 34 61 5A 23 A8 DB 62D..EB.[.a4aZ#..b
```

Signed 8 bit: 115 Signed 32 bit: 1851853427 Hexadecimal: 73
Unsigned 8 bit: 115 Unsigned 32 bit: 1851853427 Octal: 163
Signed 16 bit: 2675 Signed 64 bit: 1851853427 Binary: 01110011
Unsigned 16 bit: 2675 Unsigned 64 bit: 1851853427 Stream Length: 8
Float 32 bit: 1.741169e+28 Float 64 bit: 2.244331e+219
☑ Show little endian decoding ☐ Show unsigned and float as hexadecimal
Offset: 0x2FC4

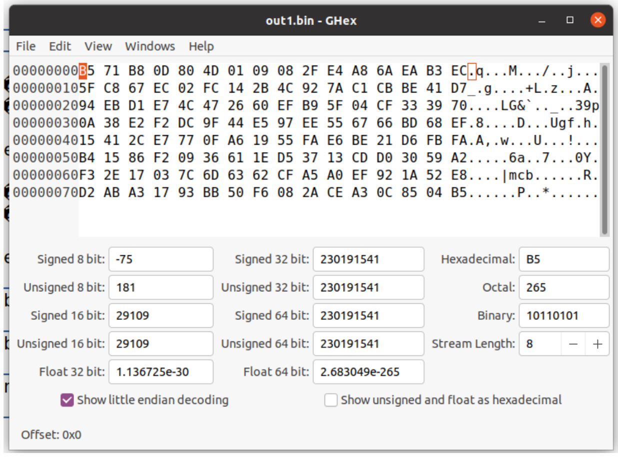
```
D\x98.EB\x05[\xe2a4a\xda#\xa8\xdbb'
```

Task 2.1

There are still differences.

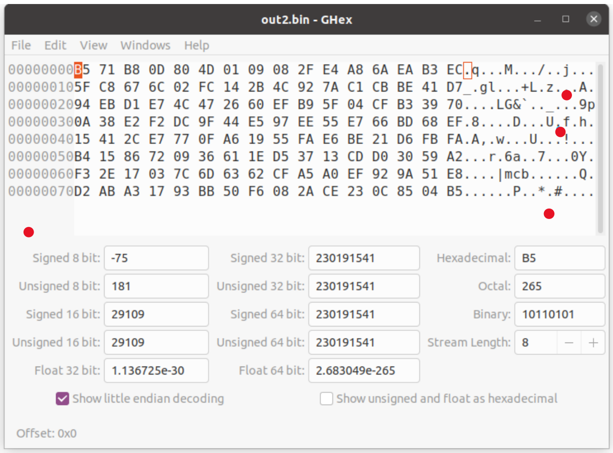


```
[04/21/23]seed@VM:~/.../d$ diff out1.bin out2.bin
Binary files out1.bin and out2.bin differ
[04/21/23]seed@VM:~/.../d$ md5sum out1.bin
b8c68d8ecf3610d7050c08c024993df8  out1.bin
[04/21/23]seed@VM:~/.../d$ md5sum ou2.bin
md5sum: ou2.bin: No such file or directory
[04/21/23]seed@VM:~/.../d$ md5sum out2.bin
b8c68d8ecf3610d7050c08c024993df8  out2.bin
[04/21/23]seed@VM:~/.../d$ md5collgen -p prefix.txt -o t1.bin t2.bin
MD5 collision generator v1.5
by Marc Stevens (http://www.win.tue.nl/hashclash/)

Using output filenames: 't1.bin' and 't2.bin'
Using prefixfile: 'prefix.txt'
Using initial value: 0630714724b14391dc74902f303d5b47

Generating first block: ...........
Generating second block: S00.........
Running time: 5.7138 s
[04/21/23]seed@VM:~/.../d$ diff out1.bin out2.bin
Binary files out1.bin and out2.bin differ
[04/21/23]seed@VM:~/.../d$ ghex out1.bin
[04/21/23]seed@VM:~/.../d$ diff -a out1.bin out2.bin
2c2
$��F���N#�k�����I�S��}�$�A�y�/�ᴀ��,
                                   m+�ǻn�cJ��a'��a����Qr]�      Q���u�_����s��펱 ��5��🄰��
\ No newline at end of file
---
$��F���N#�k����gJ�S��}�$���y�/�ᴀ��,
                                   m+�ǻn�cJ���'��a����Qr]�      Q���u�_���ms��펱 ��5��������
\ No newline at end of file
[04/21/23]seed@VM:~/.../d$
```



```
00000000B5 71 B8 0D 80 4D 01 09 08 2F E4 A8 6A EA B3 EC.q...M.../..j...
000000105F C8 67 EC 02 FC 14 2B 4C 92 7A C1 CB BE 41 D7_.g....+L.z...A.
0000002094 EB D1 E7 4C 47 26 60 EF B9 5F 04 CF 33 39 70....LG&`.._..39p
000000300A 38 E2 F2 DC 9F 44 E5 97 EE 55 67 66 BD 68 EF.8....D...Ugf.h.
0000004015 41 2C E7 77 0F A6 19 55 FA E6 BE 21 D6 FB FA.A,.w...U...!...
00000050B4 15 86 F2 09 36 61 1E D5 37 13 CD D0 30 59 A2.....6a..7...0Y.
00000060F3 2E 17 03 7C 6D 63 62 CF A5 A0 EF 92 1A 52 E8....|mcb......R.
00000070D2 AB A3 17 93 BB 50 F6 08 2A CE A3 0C 85 04 B5......P..*......
```

Signed 8 bit: -75 Signed 32 bit: 230191541 Hexadecimal: B5
Unsigned 8 bit: 181 Unsigned 32 bit: 230191541 Octal: 265
Signed 16 bit: 29109 Signed 64 bit: 230191541 Binary: 10110101
Unsigned 16 bit: 29109 Unsigned 64 bit: 230191541 Stream Length: 8
Float 32 bit: 1.136725e-30 Float 64 bit: 2.683049e-265
☑ Show little endian decoding ☐ Show unsigned and float as hexadecimal
Offset: 0x0



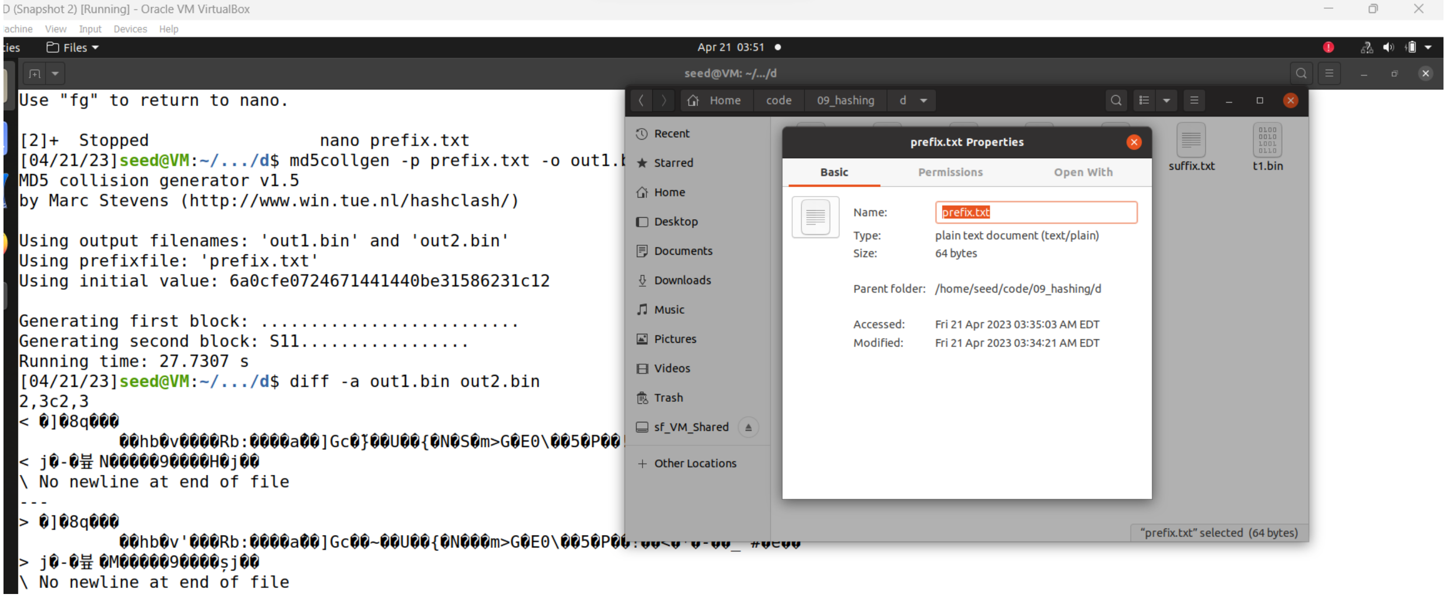
```
00000000B5 71 B8 0D 80 4D 01 09 08 2F E4 A8 6A EA B3 EC.q...M.../..j...
000000105F C8 67 6C 02 FC 14 2B 4C 92 7A C1 CB BE 41 D7_.gl...+L.z...A.
0000002094 EB D1 E7 4C 47 26 60 EF B9 5F 04 CF B3 39 70....LG&`.._...9p
000000300A 38 E2 F2 DC 9F 44 E5 97 EE 55 E7 66 BD 68 EF.8....D...U.f.h.
0000004015 41 2C E7 77 0F A6 19 55 FA E6 BE 21 D6 FB FA.A,.w...U...!...
00000050B4 15 86 72 09 36 61 1E D5 37 13 CD D0 30 59 A2...r.6a..7...0Y.
00000060F3 2E 17 03 7C 6D 63 62 CF A5 A0 EF 92 9A 51 E8....|mcb......Q.
00000070D2 AB A3 17 93 BB 50 F6 08 2A CE 23 0C 85 04 B5......P..*.#....
```

Signed 8 bit: -75 Signed 32 bit: 230191541 Hexadecimal: B5
Unsigned 8 bit: 181 Unsigned 32 bit: 230191541 Octal: 265
Signed 16 bit: 29109 Signed 64 bit: 230191541 Binary: 10110101
Unsigned 16 bit: 29109 Unsigned 64 bit: 230191541 Stream Length: 8
Float 32 bit: 1.136725e-30 Float 64 bit: 2.683049e-265
☑ Show little endian decoding ☐ Show unsigned and float as hexadecimal
Offset: 0x0

Task 2.2

Padding is added to avoid collisions. Since we are hashing, it must be a certain length to be run through the md5 hashing algorithm.

In my examples I dont see anything different when we use a 64 byte file. Down below, where I talk next, is where I just go through the demonstration again to see the difference.



```
Use "fg" to return to nano.

[2]+  Stopped                 nano prefix.txt
[04/21/23]seed@VM:~/.../d$ md5collgen -p prefix.txt -o out1.b
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: 6a0cfe0724671441440be31586231c12

Generating first block: .........................
Generating second block: S11.................
Running time: 27.7307 s
[04/21/23]seed@VM:~/.../d$ diff -a out1.bin out2.bin
2,3c2,3
< �]�8q���
         ��hb�v����Rb:����a��]Gc�}��U��{�N�S�m>G�E0\��5�P��
< j�-�븦 N�����9����H�j��
\ No newline at end of file
---
> �]�8q���
         ��hb�v'���Rb:����a��]Gc��~��U��{�N���m>G�E0\��5�P��:��<�'�-��_`#�e��
> j�-�븦 �M�����9����ș j��
\ No newline at end of file
```

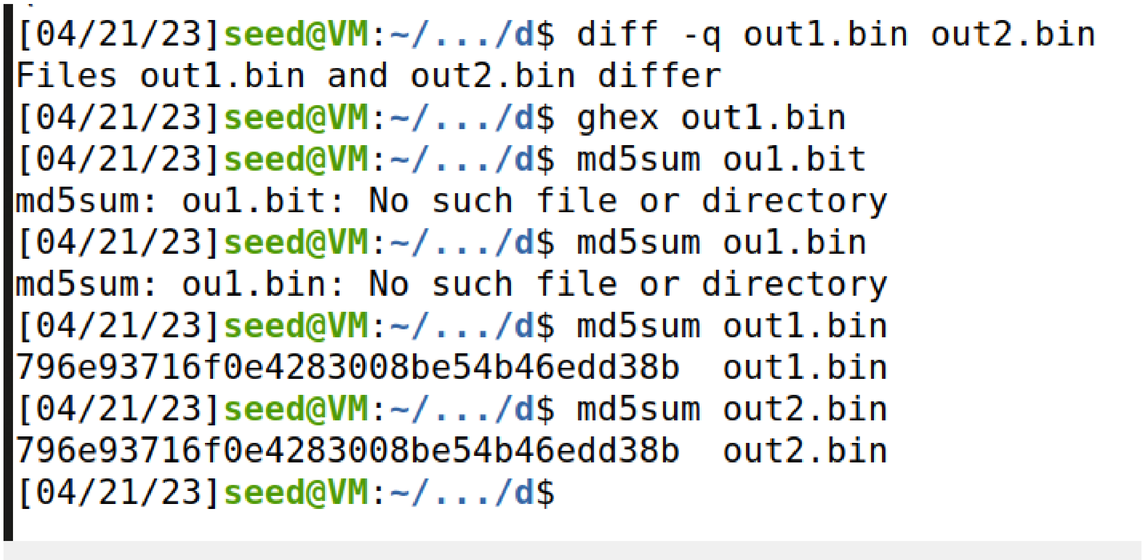
prefix.txt Properties

Name: prefix.txt
Type: plain text document (text/plain)
Size: 64 bytes

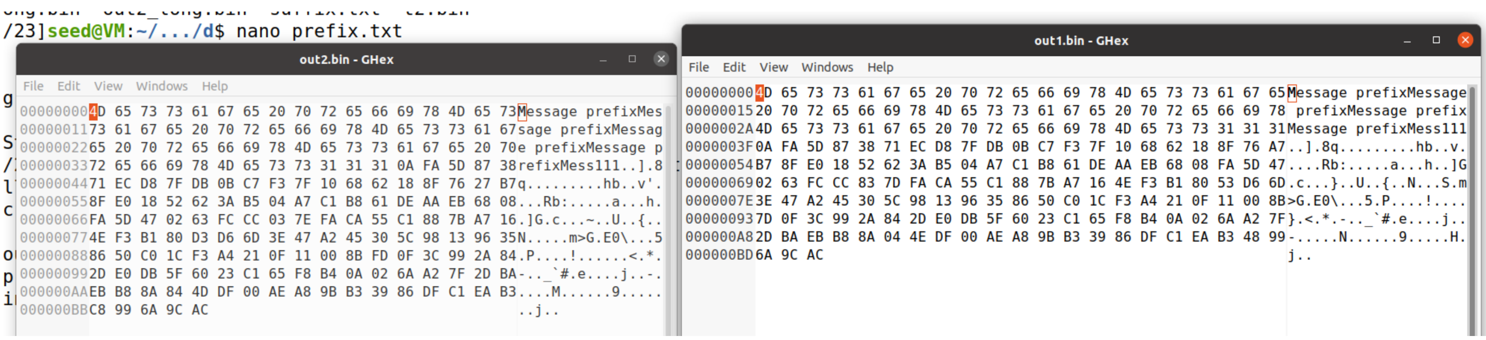
Parent folder: /home/seed/code/09_hashing/d

Accessed: Fri 21 Apr 2023 03:35:03 AM EDT
Modified: Fri 21 Apr 2023 03:34:21 AM EDT

"prefix.txt" selected (64 bytes)



```
[04/21/23]seed@VM:~/.../d$ diff -q out1.bin out2.bin
Files out1.bin and out2.bin differ
[04/21/23]seed@VM:~/.../d$ ghex out1.bin
[04/21/23]seed@VM:~/.../d$ md5sum ou1.bit
md5sum: ou1.bit: No such file or directory
[04/21/23]seed@VM:~/.../d$ md5sum ou1.bin
md5sum: ou1.bin: No such file or directory
[04/21/23]seed@VM:~/.../d$ md5sum out1.bin
796e93716f0e4283008be54b46edd38b  out1.bin
[04/21/23]seed@VM:~/.../d$ md5sum out2.bin
796e93716f0e4283008be54b46edd38b  out2.bin
[04/21/23]seed@VM:~/.../d$
```



```
ong.bin  out2_long.bin  suffix.txt  t2.bin
/23]seed@VM:~/.../d$ nano prefix.txt
```

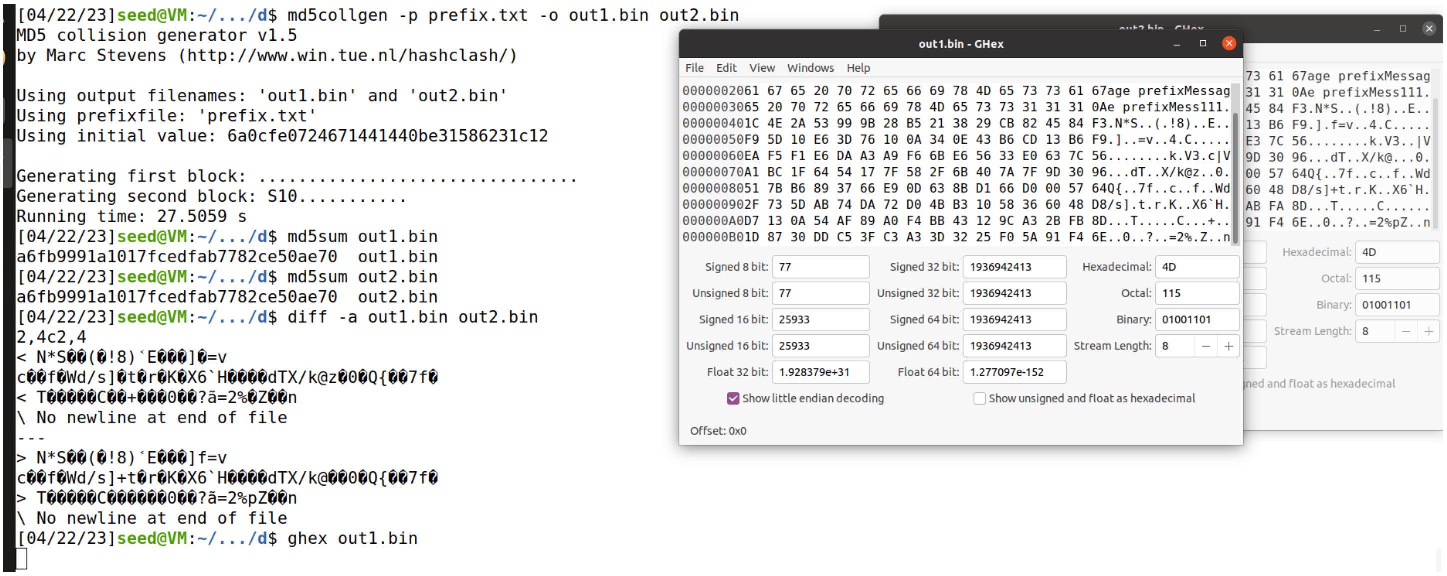
out2.bin - GHex

```
000000004D 65 73 73 61 67 65 20 70 72 65 66 69 78 4D 65 73Message prefixMes
0000001173 61 67 65 20 70 72 65 66 69 78 4D 65 73 73 61 67sage prefixMessag
0000002265 20 70 72 65 66 69 78 4D 65 73 73 61 67 65 20 70e prefixMessage p
0000003372 65 66 69 78 4D 65 73 73 31 31 31 0A FA 5D 87 38refixMess111..].8
0000004471 EC D8 7F DB 0B C7 F3 7F 10 68 62 18 8F 76 27 B7q.........hb..v'.
000000558F E0 18 52 62 3A B5 04 A7 C1 B8 61 DE AA EB 68 08...Rb:......a...h.
00000066FA 5D 47 02 63 FC CC 03 7E FA CA 55 C1 88 7B A7 16.]G.c...~..U..{..
000000774E F3 B1 80 D3 D6 6D 3E 47 A2 45 30 5C 98 13 96 35N.....m>G.E0\...5
0000008886 50 C0 1C F3 A4 21 0F 11 00 8B FD 0F 3C 99 2A 84.P....!......<.*.
000000992D E0 DB 5F 60 23 C1 65 F8 B4 0A 02 6A A2 7F 2D BA-.._`#.e....j..-.
000000AAEB B8 8A 84 4D DF 00 AE A8 9B B3 39 86 DF C1 EA B3....M......9.....
000000BBC8 99 6A 9C AC                                    ..j..
```

out1.bin - GHex

```
000000004D 65 73 73 61 67 65 20 70 72 65 66 69 78 4D 65 73 73 61 67 65Message prefixMessage
0000001520 70 72 65 66 69 78 4D 65 73 73 61 67 65 20 70 72 65 66 69 78 prefixMessage prefix
0000002A4D 65 73 73 61 67 65 20 70 72 65 66 69 78 4D 65 73 73 31 31 31Message prefixMess111
0000003F0A FA 5D 87 38 71 EC D8 7F DB 0B C7 F3 7F 10 68 62 18 8F 76 A7..].8q.........hb..v.
00000054B7 8F E0 18 52 62 3A B5 04 A7 C1 B8 61 DE AA EB 68 08 FA 5D 47....Rb:.....a...h..]G
0000006902 63 FC CC 83 7D FA CA 55 C1 88 7B A7 16 4E F3 B1 80 53 D6 6D.c...}..U..{..N...S.m
0000007E3E 47 A2 45 30 5C 98 13 96 35 86 50 C0 1C F3 A4 21 0F 11 00 8B>G.E0\...5.P....!....
000000937D 0F 3C 99 2A 84 2D E0 DB 5F 60 23 C1 65 F8 B4 0A 02 6A A2 7F}.<.*.-.._`#.e....j..
000000A82D BA EB B8 8A 04 4E DF 00 AE A8 9B B3 39 86 DF C1 EA B3 48 99-.....N......9.....H.
000000BD6A 9C AC                                                      j..
```

Here is where I redo the demonstration, this is with a 15kB file.



```
[04/22/23]seed@VM:~/.../d$ 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: 6a0cfe0724671441440be31586231c12

Generating first block: ...............................
Generating second block: S10...........
Running time: 27.5059 s
[04/22/23]seed@VM:~/.../d$ md5sum out1.bin
a6fb9991a1017fcedfab7782ce50ae70  out1.bin
[04/22/23]seed@VM:~/.../d$ md5sum out2.bin
a6fb9991a1017fcedfab7782ce50ae70  out2.bin
[04/22/23]seed@VM:~/.../d$ diff -a out1.bin out2.bin
2,4c2,4
< N*S��(�!8)`E���]�=v
c��f�Wd/s]�t�r�K�X6`H����dTX/k@z�0�Q{��7f�
< T�����C��+���0��?ã=2%�Z��n
\ No newline at end of file
---
> N*S��(�!8)`E���]f=v
c��f�Wd/s]+t�r�K�X6`H����dTX/k@���0�Q{��7f�
> T�����C�������0��?ã=2%pZ��n
\ No newline at end of file
[04/22/23]seed@VM:~/.../d$ ghex out1.bin
```

out1.bin - GHex

File Edit View Windows Help

```
0000002061 67 65 20 70 72 65 66 69 78 4D 65 73 73 61 67age prefixMessag
0000003065 20 70 72 65 66 69 78 4D 65 73 73 31 31 31 0Ae prefixMess111.
000000401C 4E 2A 53 99 9B 28 B5 21 38 29 CB 82 45 84 F3.N*S..(.!8)..E..
00000050F9 5D 10 E6 3D 76 10 0A 34 0E 43 B6 CD 13 B6 F9.]..=v..4.C.....
00000060EA F5 F1 E6 DA A3 A9 F6 6B E6 56 33 E0 63 7C 56........k.V3.c|V
00000070A1 BC 1F 64 54 17 7F 58 2F 6B 40 7A 7F 9D 30 96...dT..X/k@z..0.
0000008051 7B B6 89 37 66 E9 0D 63 8B D1 66 D0 00 57 64Q{..7f..c..f..Wd
000000902F 73 5D AB 74 DA 72 D0 4B B3 10 58 36 60 48 D8/s].t.r.K..X6`H.
000000A0D7 13 0A 54 AF 89 A0 F4 BB 43 12 9C A3 2B FB 8D...T.....C...+..
000000B01D 87 30 DD C5 3F C3 A3 3D 32 25 F0 5A 91 F4 6E..0..?..=2%.Z..n
```

Signed 8 bit: 77 | Signed 32 bit: 1936942413 | Hexadecimal: 4D
Unsigned 8 bit: 77 | Unsigned 32 bit: 1936942413 | Octal: 115
Signed 16 bit: 25933 | Signed 64 bit: 1936942413 | Binary: 01001101
Unsigned 16 bit: 25933 | Unsigned 64 bit: 1936942413 | Stream Length: 8
Float 32 bit: 1.928379e+31 | Float 64 bit: 1.277097e-152

☑ Show little endian decoding ☐ Show unsigned and float as hexadecimal

Offset: 0x0

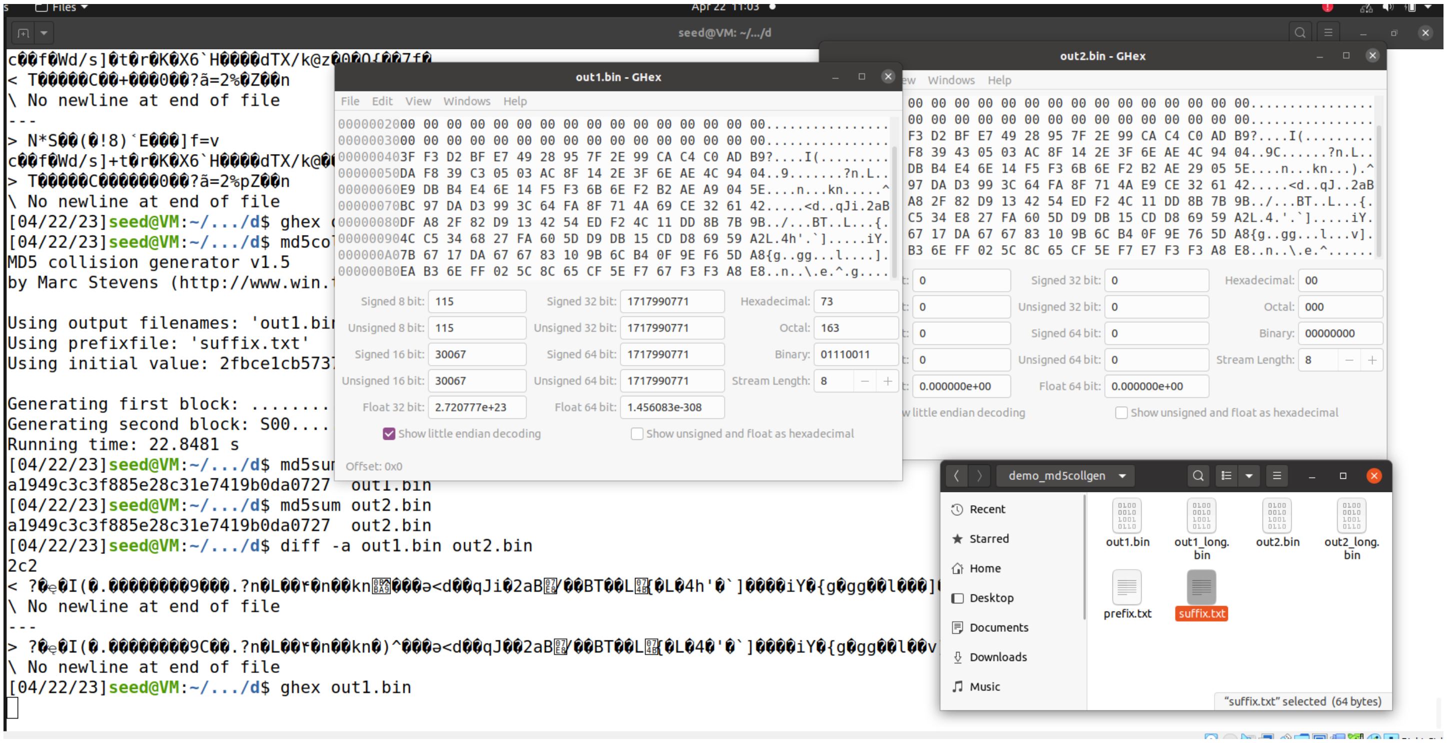
out2.bin - GHex

```
73 61 67age prefixMessag
31 31 0Ae prefixMess111.
45 84 F3.N*S..(.!8)..E..
13 B6 F9.].f=v..4.C.....
E3 7C 56........k.V3..|V
9D 30 96...dT..X/k@...0.
00 57 64Q{..7f..c..f..Wd
60 48 D8/s]+t.r.K..X6`H.
AB FA 8D...T.....C......
91 F4 6E..0..?..=2%pZ..n
```

Hexadecimal: 4D
Octal: 115
Binary: 01001101
Stream Length: 8

ned and float as hexadecimal

Here is with a 64kb file.



```
c��f�Wd/s]�t�r�K�X6`H����dTX/k@z�0�Q{��7f�
< T�����C��+���0��?ã=2%�Z��n
\ No newline at end of file
---
> N*S��(�!8)`E���]f=v
c��f�Wd/s]+t�r�K�X6`H����dTX/k@�
> T�����C�������0��?ã=2%pZ��n
\ No newline at end of file
[04/22/23]seed@VM:~/.../d$ ghex
[04/22/23]seed@VM:~/.../d$ md5co
MD5 collision generator v1.5
by Marc Stevens (http://www.win.
Using output filenames: 'out1.bi
Using prefixfile: 'suffix.txt'
Using initial value: 2fbce1cb573

Generating first block: ........
Generating second block: S00....
Running time: 22.8481 s
[04/22/23]seed@VM:~/.../d$ md5su
a1949c3c3f885e28c31e7419b0da0727  out1.bin
[04/22/23]seed@VM:~/.../d$ md5sum out2.bin
a1949c3c3f885e28c31e7419b0da0727  out2.bin
[04/22/23]seed@VM:~/.../d$ diff -a out1.bin out2.bin
2c2
< ?�ｅ�I(�.���������9���.?n�L��r�n��kn�����ə<d��qJi�2aB�/��BT��L�[�L�4h'�`]����iY�{g�gg��l���]
\ No newline at end of file
---
> ?�ｅ�I(�.��������9C��.?n�L��r�n��kn�)^���ə<d��qJ��2aB�/��BT��L�[�L�4�'�`]����iY�{g�gg��l��v
\ No newline at end of file
[04/22/23]seed@VM:~/.../d$ ghex out1.bin
```

out1.bin - GHex

File Edit View Windows Help

```
0000002000 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00................
0000003000 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00................
000000403F F3 D2 BF E7 49 28 95 7F 2E 99 CA C4 C0 AD B9?....I(.........
00000050DA F8 39 C3 05 03 AC 8F 14 2E 3F 6E AE 4C 94 04..9.......?n.L..
00000060E9 DB B4 E4 6E 14 F5 F3 6B 6E F2 B2 AE A9 04 5E....n...kn.....^
00000070BC 97 DA D3 99 3C 64 FA 8F 71 4A 69 CE 32 61 42.....<d..qJi.2aB
00000080DF A8 2F 82 D9 13 42 54 ED F2 4C 11 DD 8B 7B 9B../...BT..L...{.
000000904C C5 34 68 27 FA 60 5D D9 DB 15 CD D8 69 59 A2L.4h'.`].....iY.
000000A07B 67 17 DA 67 67 83 10 9B 6C B4 0F 9E F6 5D A8{g..gg...l....].
000000B0EA B3 6E FF 02 5C 8C 65 CF 5E F7 67 F3 F3 A8 E8..n..\.e.^.g....
```

Signed 8 bit: 115 | Signed 32 bit: 1717990771 | Hexadecimal: 73
Unsigned 8 bit: 115 | Unsigned 32 bit: 1717990771 | Octal: 163
Signed 16 bit: 30067 | Signed 64 bit: 1717990771 | Binary: 01110011
Unsigned 16 bit: 30067 | Unsigned 64 bit: 1717990771 | Stream Length: 8
Float 32 bit: 2.720777e+23 | Float 64 bit: 1.456083e-308

☑ Show little endian decoding ☐ Show unsigned and float as hexadecimal

Offset: 0x0

out2.bin - GHex

ew Windows Help

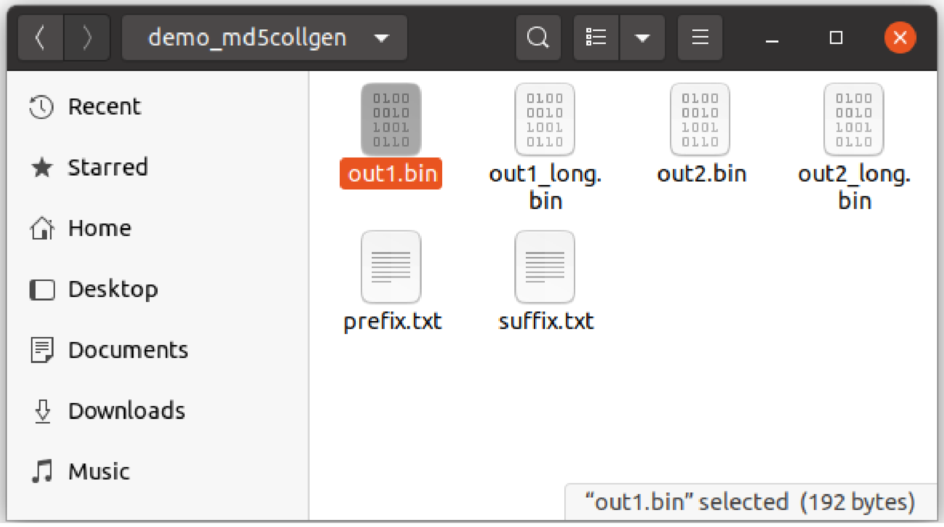
```
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00................
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00................
F3 D2 BF E7 49 28 95 7F 2E 99 CA C4 C0 AD B9?....I(.........
F8 39 43 05 03 AC 8F 14 2E 3F 6E AE 4C 94 04..9C......?n.L..
DB B4 E4 6E 14 F5 F3 6B 6E F2 B2 AE 29 05 5E....n...kn...).^
97 DA D3 99 3C 64 FA 8F 71 4A E9 CE 32 61 42.....<d..qJ..2aB
A8 2F 82 D9 13 42 54 ED F2 4C 11 DD 8B 7B 9B../...BT..L...{.
C5 34 E8 27 FA 60 5D D9 DB 15 CD D8 69 59 A2L.4.'.`].....iY.
67 17 DA 67 67 83 10 9B 6C B4 0F 9E 76 5D A8{g..gg...l...v].
B3 6E FF 02 5C 8C 65 CF 5E F7 E7 F3 F3 A8 E8..n..\.e.^......
```

t: 0 | Signed 32 bit: 0 | Hexadecimal: 00
t: 0 | Unsigned 32 bit: 0 | Octal: 000
t: 0 | Signed 64 bit: 0 | Binary: 00000000
t: 0 | Unsigned 64 bit: 0 | Stream Length: 8
t: 0.000000e+00 | Float 64 bit: 0.000000e+00

w little endian decoding ☐ Show unsigned and float as hexadecimal

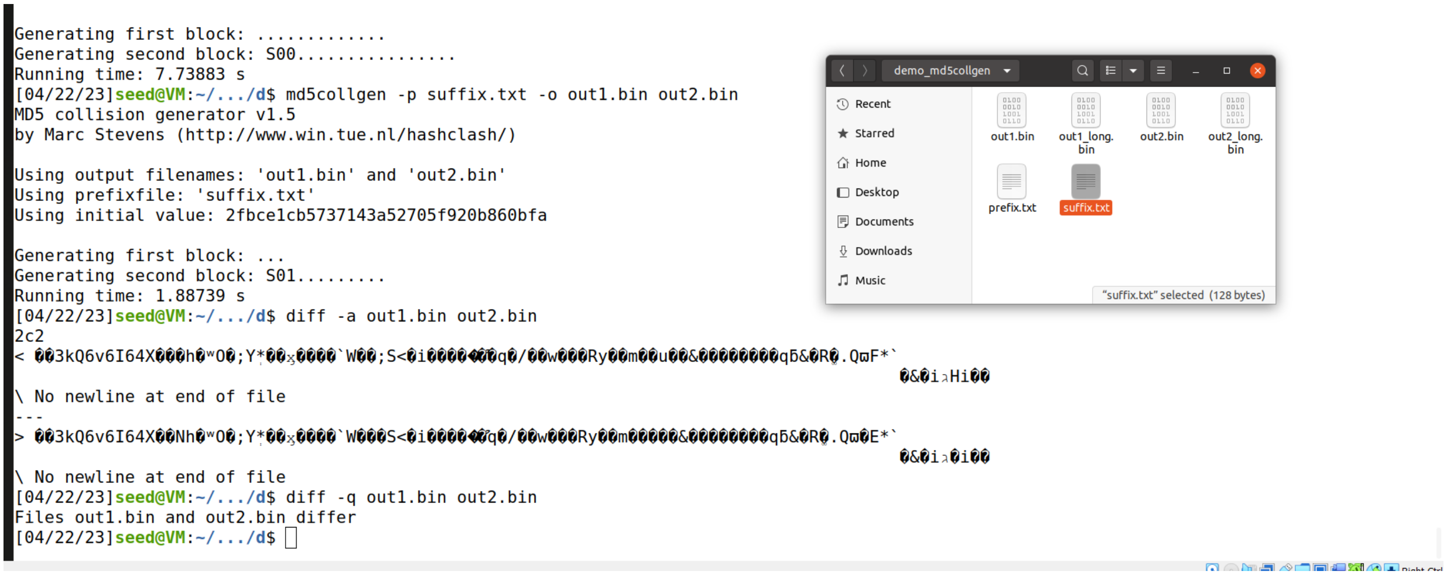
I think the answer is supposed to be that there is a difference between the two, and that's because a 64kB file does not add padding as its not required since it's a power of 2.

Looking at the size of the out1.bin when we hashed it using a 64kB file vs when we look at the size of the out1.bin. It shows the same size, which is interesting, this could be proof of it adding padding and then hashing.



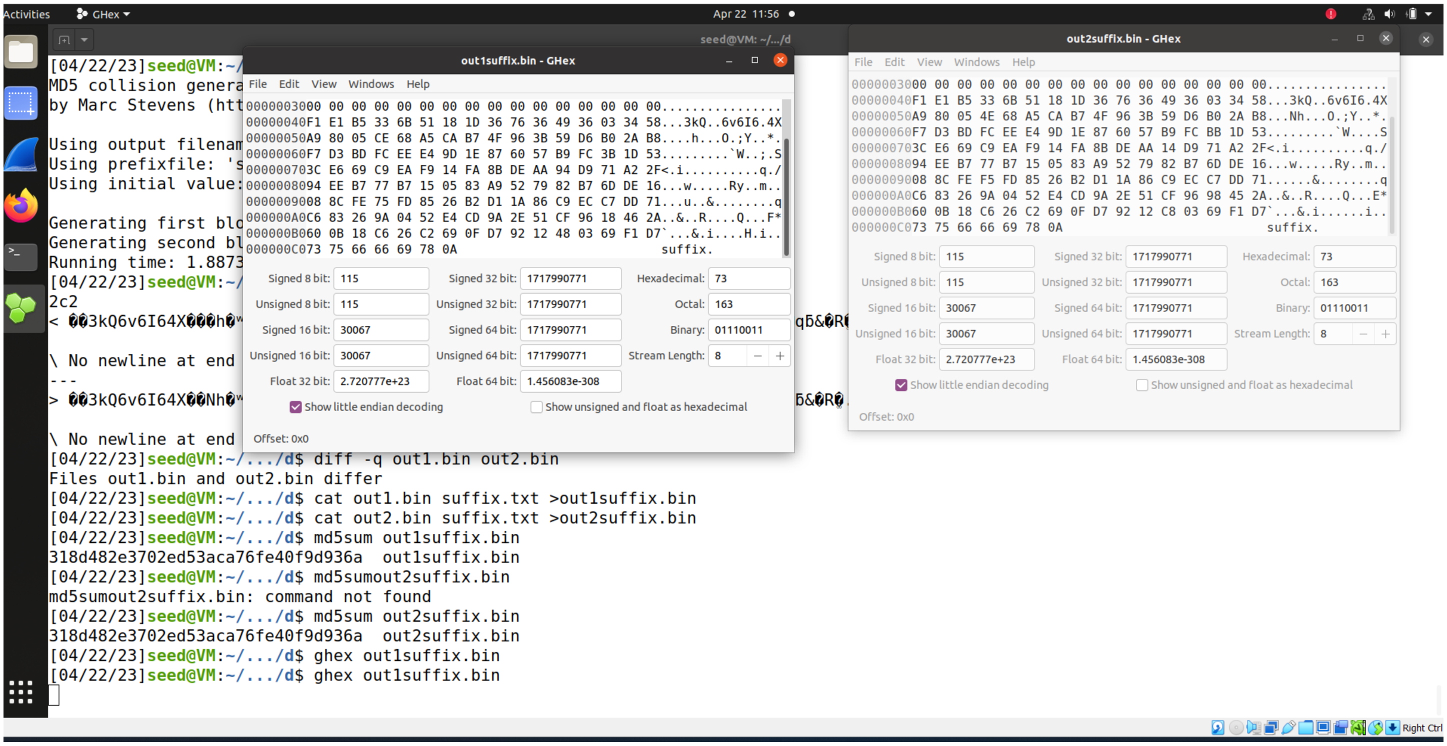
Task 2.4

“Technically” in my experiments, they are not the same once hashed, even though when we go to unhash them they will be the same. I ran these experiments for about 3 hours trying to see if there truly was a difference between a hashed 64kB with two outputs and same hash, I could never get it to give me the exact thing. Maybe that’s the point? But I don’t see how or why the out1.bin and out2.bin would be different. But to answer the question for 2.4, it would be different. The two output files would always be different. Just to prove my lab I changed the size of the input file to 128kB and checked for a difference.



```
Generating first block: .............
Generating second block: S00................
Running time: 7.73883 s
[04/22/23]seed@VM:~/.../d$ md5collgen -p suffix.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: 'suffix.txt'
Using initial value: 2fbce1cb5737143a52705f920b860bfa

Generating first block: ...
Generating second block: S01.........
Running time: 1.88739 s
[04/22/23]seed@VM:~/.../d$ diff -a out1.bin out2.bin
2c2
< ��3kQ6v6I64X���h�ʷO�;Y*��ᵪ����`W��;S<�i����ǔ�q�/��w���Ry��m��u��&���������qƃ&�R�.QϖF*`
                                                                                    �&�iᲹHi��
\ No newline at end of file
---
> ��3kQ6v6I64X��Nh�ʷO�;Y*��ᵪ����`W���S<�i����ǔ�q�/��w���Ry��m�����&���������qƃ&�R�.Qϖ�E*`
                                                                                    �&�iᲹ�i��
\ No newline at end of file
[04/22/23]seed@VM:~/.../d$ diff -q out1.bin out2.bin
Files out1.bin and out2.bin differ
[04/22/23]seed@VM:~/.../d$ 
```

Task 3



```
[04/22/23]seed@VM:~/
MD5 collision genera
by Marc Stevens (htt

Using output filenam
Using prefixfile: 's
Using initial value:

Generating first blo
Generating second bl
Running time: 1.8873
[04/22/23]seed@VM:~/
2c2
< ��3kQ6v6I64X���h�ʷ                                          qЂ&�R�

\ No newline at end
---
> ��3kQ6v6I64X��Nh�ʷ                                          Ђ&�R�.

\ No newline at end
[04/22/23]seed@VM:~/.../d$ diff -q out1.bin out2.bin
Files out1.bin and out2.bin differ
[04/22/23]seed@VM:~/.../d$ cat out1.bin suffix.txt >out1suffix.bin
[04/22/23]seed@VM:~/.../d$ cat out2.bin suffix.txt >out2suffix.bin
[04/22/23]seed@VM:~/.../d$ md5sum out1suffix.bin
318d482e3702ed53aca76fe40f9d936a  out1suffix.bin
[04/22/23]seed@VM:~/.../d$ md5sumout2suffix.bin
md5sumout2suffix.bin: command not found
[04/22/23]seed@VM:~/.../d$ md5sum out2suffix.bin
318d482e3702ed53aca76fe40f9d936a  out2suffix.bin
[04/22/23]seed@VM:~/.../d$ ghex out1suffix.bin
[04/22/23]seed@VM:~/.../d$ ghex out1suffix.bin
```

out1suffix.bin - GHex

```
0000003000 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00................
00000040F1 E1 B5 33 6B 51 18 1D 36 76 36 49 36 03 34 58...3kQ..6v6I6.4X
00000050A9 80 05 CE 68 A5 CA B7 4F 96 3B 59 D6 B0 2A B8....h...O.;Y..*.
00000060F7 D3 BD FC EE E4 9D 1E 87 60 57 B9 FC 3B 1D 53.........`W..;.S
000000703C E6 69 C9 EA F9 14 FA 8B DE AA 94 D9 71 A2 2F<.i.........q./
0000008094 EE B7 77 B7 15 05 83 A9 52 79 82 B7 6D DE 16...w.....Ry..m..
0000009008 8C FE 75 FD 85 26 B2 D1 1A 86 C9 EC C7 DD 71...u..&........q
000000A0C6 83 26 9A 04 52 E4 CD 9A 2E 51 CF 96 18 46 2A..&..R....Q...F*
000000B060 0B 18 C6 26 C2 69 0F D7 92 12 48 03 69 F1 D7`...&.i....H.i..
000000C073 75 66 66 69 78 0A                              suffix.
```

Signed 8 bit:	115	Signed 32 bit:	1717990771	Hexadecimal:	73
Unsigned 8 bit:	115	Unsigned 32 bit:	1717990771	Octal:	163
Signed 16 bit:	30067	Signed 64 bit:	1717990771	Binary:	01110011
Unsigned 16 bit:	30067	Unsigned 64 bit:	1717990771	Stream Length:	8
Float 32 bit:	2.720777e+23	Float 64 bit:	1.456083e-308		

☑ Show little endian decoding ☐ Show unsigned and float as hexadecimal

Offset: 0x0

out2suffix.bin - GHex

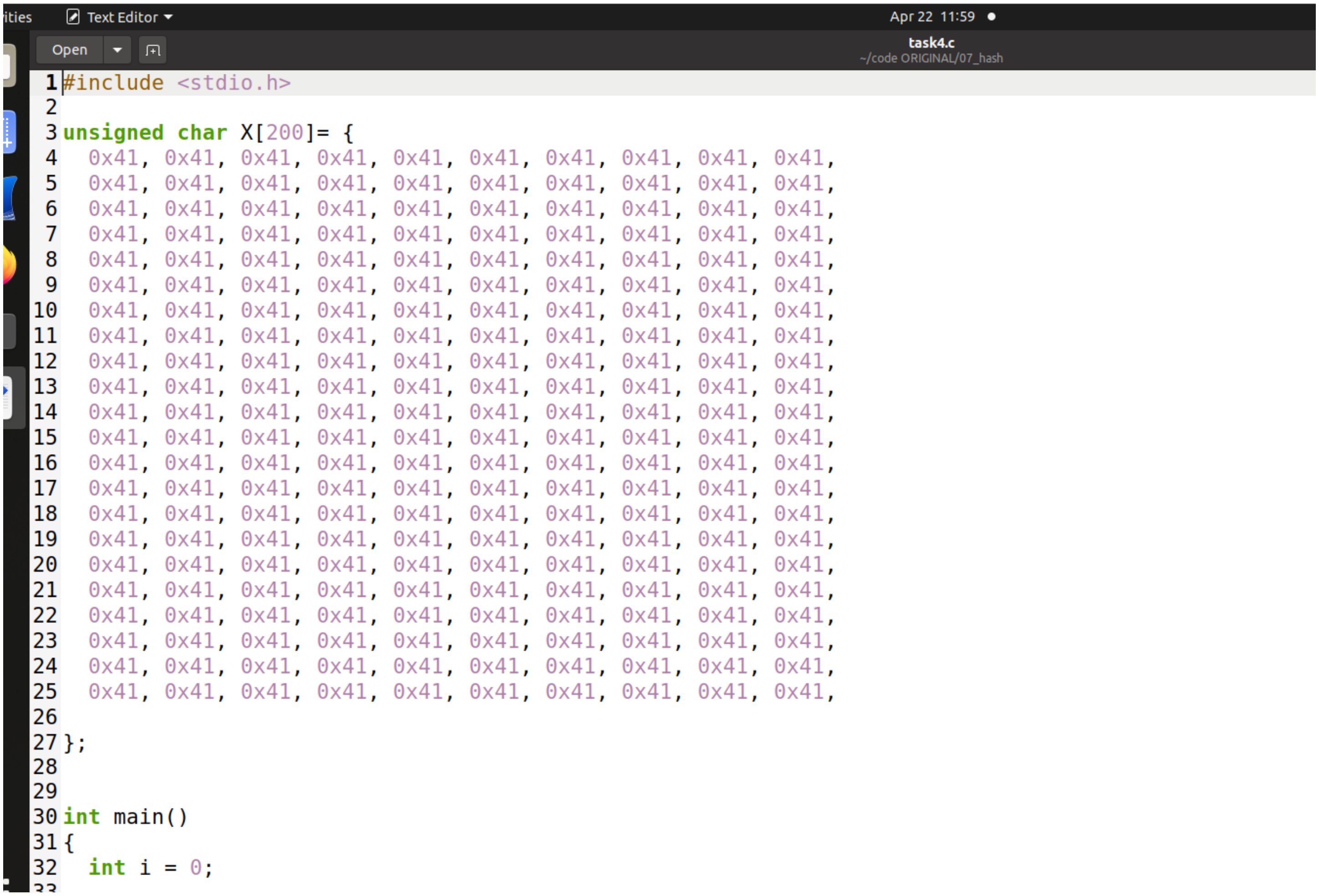
```
0000003000 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00................
00000040F1 E1 B5 33 6B 51 18 1D 36 76 36 49 36 03 34 58...3kQ..6v6I6.4X
00000050A9 80 05 4E 68 A5 CA B7 4F 96 3B 59 D6 B0 2A B8...Nh...O.;Y..*.
00000060F7 D3 BD FC EE E4 9D 1E 87 60 57 B9 FC BB 1D 53.........`W....S
000000703C E6 69 C9 EA F9 14 FA 8B DE AA 14 D9 71 A2 2F<.i.........q./
0000008094 EE B7 77 B7 15 05 83 A9 52 79 82 B7 6D DE 16...w.....Ry..m..
0000009008 8C FE F5 FD 85 26 B2 D1 1A 86 C9 EC C7 DD 71......&........q
000000A0C6 83 26 9A 04 52 E4 CD 9A 2E 51 CF 96 98 45 2A..&..R....Q...E*
000000B060 0B 18 C6 26 C2 69 0F D7 92 12 C8 03 69 F1 D7`...&.i......i..
000000C073 75 66 66 69 78 0A                              suffix.
```

Signed 8 bit:	115	Signed 32 bit:	1717990771	Hexadecimal:	73
Unsigned 8 bit:	115	Unsigned 32 bit:	1717990771	Octal:	163
Signed 16 bit:	30067	Signed 64 bit:	1717990771	Binary:	01110011
Unsigned 16 bit:	30067	Unsigned 64 bit:	1717990771	Stream Length:	8
Float 32 bit:	2.720777e+23	Float 64 bit:	1.456083e-308		

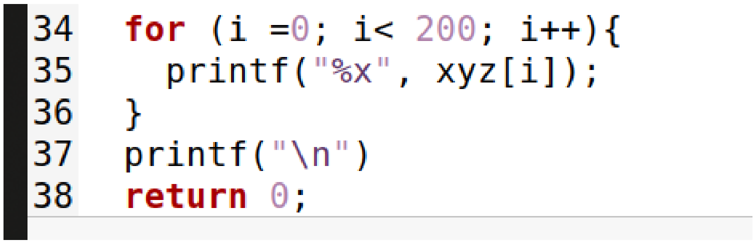
☑ Show little endian decoding ☐ Show unsigned and float as hexadecimal

Offset: 0x0

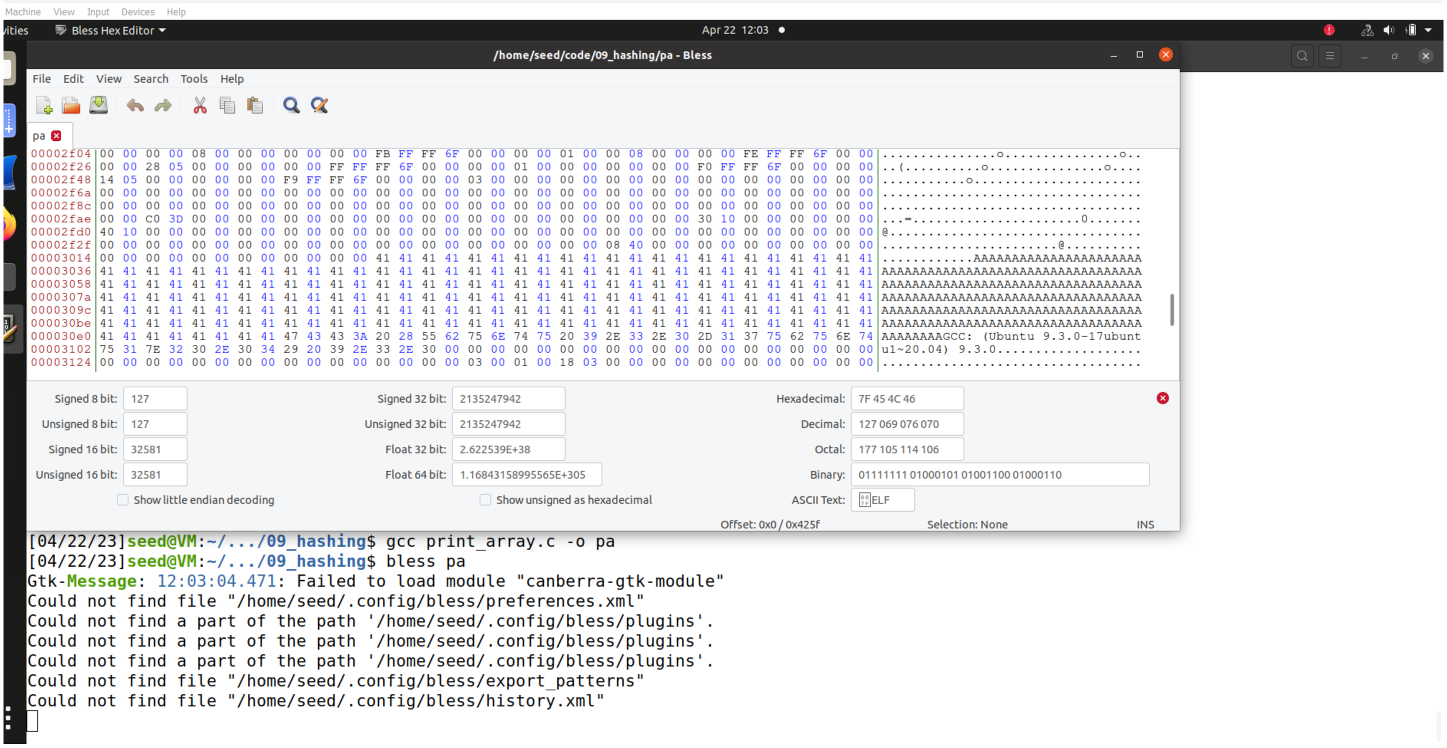
Task 4



```
#include <stdio.h>

unsigned char X[200]= {
  0x41, 0x41, 0x41, 0x41, 0x41, 0x41, 0x41, 0x41, 0x41, 0x41,
  0x41, 0x41, 0x41, 0x41, 0x41, 0x41, 0x41, 0x41, 0x41, 0x41,
  0x41, 0x41, 0x41, 0x41, 0x41, 0x41, 0x41, 0x41, 0x41, 0x41,
  0x41, 0x41, 0x41, 0x41, 0x41, 0x41, 0x41, 0x41, 0x41, 0x41,
  0x41, 0x41, 0x41, 0x41, 0x41, 0x41, 0x41, 0x41, 0x41, 0x41,
  0x41, 0x41, 0x41, 0x41, 0x41, 0x41, 0x41, 0x41, 0x41, 0x41,
  0x41, 0x41, 0x41, 0x41, 0x41, 0x41, 0x41, 0x41, 0x41, 0x41,
  0x41, 0x41, 0x41, 0x41, 0x41, 0x41, 0x41, 0x41, 0x41, 0x41,
  0x41, 0x41, 0x41, 0x41, 0x41, 0x41, 0x41, 0x41, 0x41, 0x41,
  0x41, 0x41, 0x41, 0x41, 0x41, 0x41, 0x41, 0x41, 0x41, 0x41,
  0x41, 0x41, 0x41, 0x41, 0x41, 0x41, 0x41, 0x41, 0x41, 0x41,
  0x41, 0x41, 0x41, 0x41, 0x41, 0x41, 0x41, 0x41, 0x41, 0x41,
  0x41, 0x41, 0x41, 0x41, 0x41, 0x41, 0x41, 0x41, 0x41, 0x41,
  0x41, 0x41, 0x41, 0x41, 0x41, 0x41, 0x41, 0x41, 0x41, 0x41,
  0x41, 0x41, 0x41, 0x41, 0x41, 0x41, 0x41, 0x41, 0x41, 0x41,
  0x41, 0x41, 0x41, 0x41, 0x41, 0x41, 0x41, 0x41, 0x41, 0x41,
  0x41, 0x41, 0x41, 0x41, 0x41, 0x41, 0x41, 0x41, 0x41, 0x41,
  0x41, 0x41, 0x41, 0x41, 0x41, 0x41, 0x41, 0x41, 0x41, 0x41,
  0x41, 0x41, 0x41, 0x41, 0x41, 0x41, 0x41, 0x41, 0x41, 0x41,
  0x41, 0x41, 0x41, 0x41, 0x41, 0x41, 0x41, 0x41, 0x41, 0x41,
  0x41, 0x41, 0x41, 0x41, 0x41, 0x41, 0x41, 0x41, 0x41, 0x41,
  0x41, 0x41, 0x41, 0x41, 0x41, 0x41, 0x41, 0x41, 0x41, 0x41,

};


int main()
{
  int i = 0;
```



```
for (i =0; i< 200; i++){
  printf("%x", xyz[i]);
}
printf("\n")
return 0;
```



Machine View Input Devices Help
ities
Bless Hex Editor
Apr 22 12:03
/home/seed/code/09_hashing/pa - Bless
File Edit View Search Tools Help
pa

```
00002f04 00 00 00 00 08 00 00 00 00 00 00 00 FB FF FF 6F 00 00 00 00 01 00 00 08 00 00 00 00 FE FF FF 6F 00 00 ...............o...............o..
00002f26 00 00 28 05 00 00 00 00 00 00 FF FF FF 6F 00 00 00 00 01 00 00 00 00 00 00 00 F0 FF FF 6F 00 00 00 00 ..(..........o...............o....
00002f48 14 05 00 00 00 00 00 00 F9 FF FF 6F 00 00 00 00 03 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ...........o......................
00002f6a 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 ..................................
00002f8c 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 ..................................
00002fae 00 00 C0 3D 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 30 10 00 00 00 00 00 00 ...=......................0.......
00002fd0 40 10 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 @.................................
00002f2f 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 08 40 00 00 00 00 00 00 00 00 00 00 .......................@..........
00003014 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 41 41 41 41 41 41 ............AAAAAAAAAAAAAAAAAAAAAA
00003036 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
00003058 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
0000307a 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
0000309c 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
000030be 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
000030e0 41 41 41 41 41 41 41 41 47 43 43 3A 20 28 55 62 75 6E 74 75 20 39 2E 33 2E 30 2D 31 37 75 62 75 6E 74 AAAAAAAAGCC: (Ubuntu 9.3.0-17ubunt
00003102 75 31 7E 32 30 2E 30 34 29 20 39 2E 33 2E 30 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 u1~20.04) 9.3.0...................
00003124 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 03 00 01 00 18 03 00 00 00 00 00 00 00 00 00 00 00 00 ..................................
```

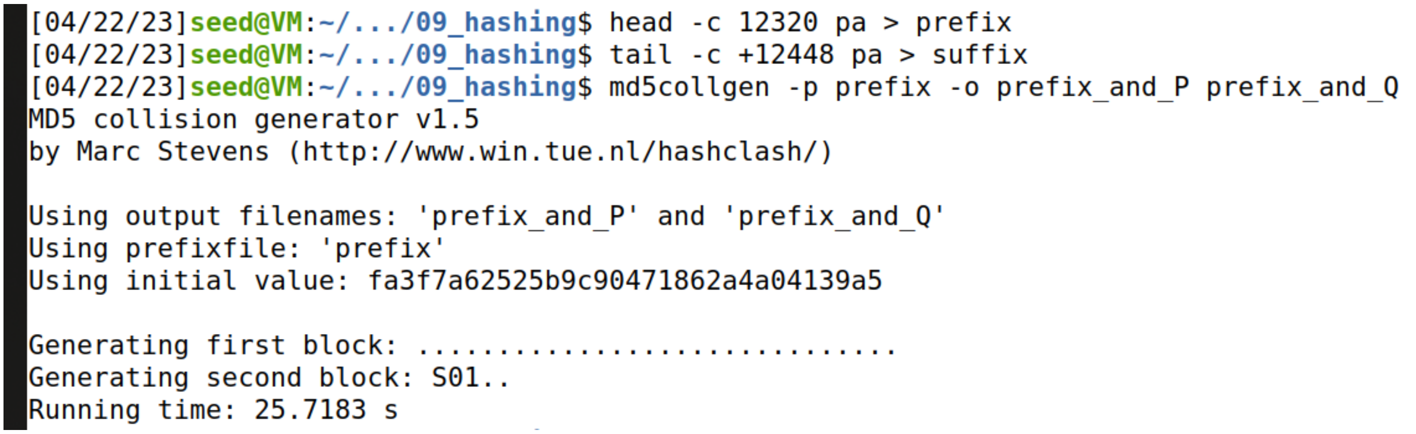
Signed 8 bit: 127
Unsigned 8 bit: 127
Signed 16 bit: 32581
Unsigned 16 bit: 32581
□ Show little endian decoding

Signed 32 bit: 2135247942
Unsigned 32 bit: 2135247942
Float 32 bit: 2.622539E+38
Float 64 bit: 1.16843158995565E+305
□ Show unsigned as hexadecimal

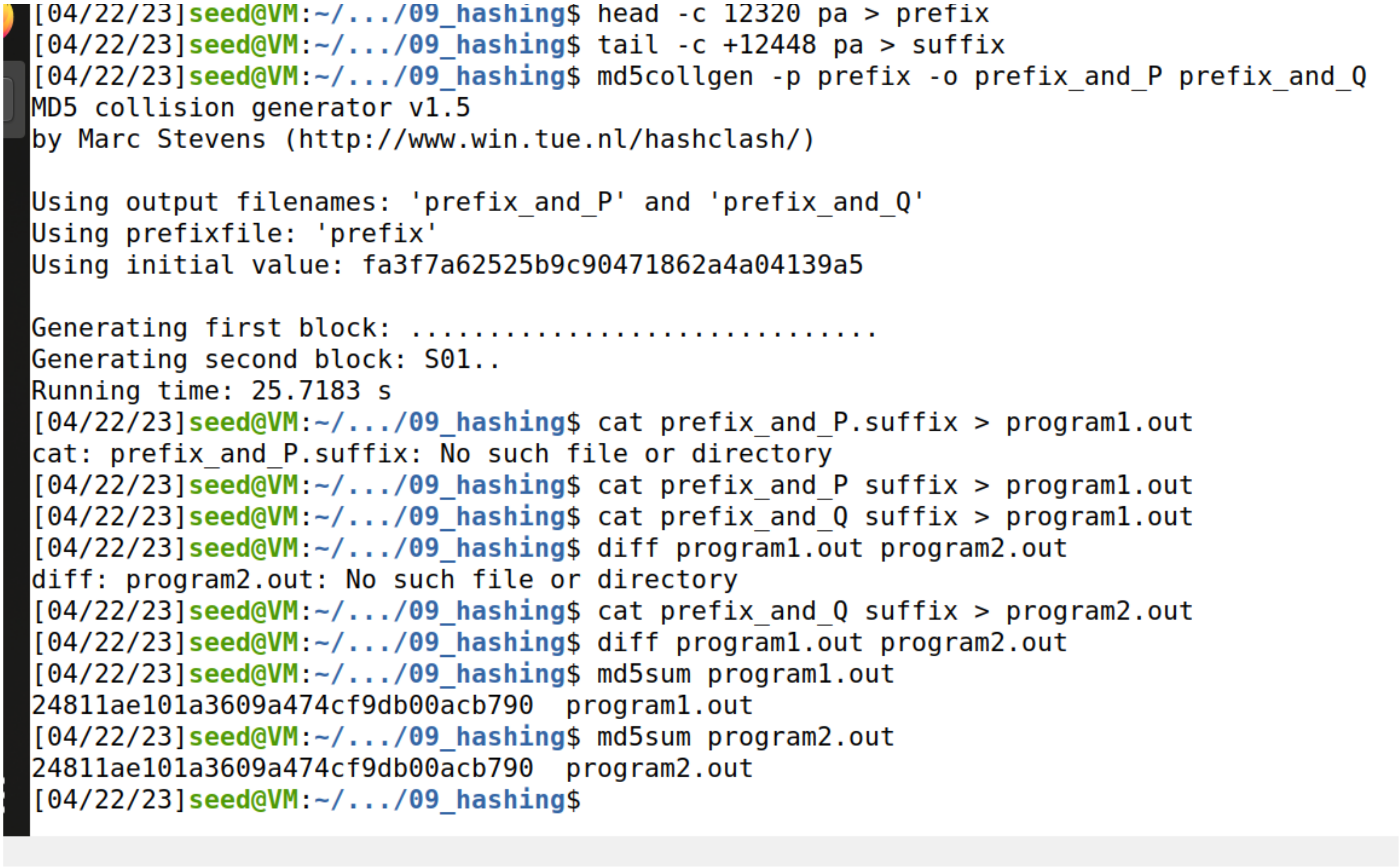
Hexadecimal: 7F 45 4C 46
Decimal: 127 069 076 070
Octal: 177 105 114 106
Binary: 01111111 01000101 01001100 01000110
ASCII Text: ELF

Offset: 0x0 / 0x425f   Selection: None   INS

```
[04/22/23]seed@VM:~/.../09_hashing$ gcc print_array.c -o pa
[04/22/23]seed@VM:~/.../09_hashing$ bless pa
Gtk-Message: 12:03:04.471: Failed to load module "canberra-gtk-module"
Could not find file "/home/seed/.config/bless/preferences.xml"
Could not find a part of the path '/home/seed/.config/bless/plugins'.
Could not find a part of the path '/home/seed/.config/bless/plugins'.
Could not find a part of the path '/home/seed/.config/bless/plugins'.
Could not find file "/home/seed/.config/bless/export_patterns"
Could not find file "/home/seed/.config/bless/history.xml"
```



```
[04/22/23]seed@VM:~/.../09_hashing$ head -c 12320 pa > prefix
[04/22/23]seed@VM:~/.../09_hashing$ tail -c +12448 pa > suffix
[04/22/23]seed@VM:~/.../09_hashing$ md5collgen -p prefix -o prefix_and_P prefix_and_Q
MD5 collision generator v1.5
by Marc Stevens (http://www.win.tue.nl/hashclash/)

Using output filenames: 'prefix_and_P' and 'prefix_and_Q'
Using prefixfile: 'prefix'
Using initial value: fa3f7a62525b9c90471862a4a04139a5

Generating first block: ..............................
Generating second block: S01..
Running time: 25.7183 s
```



```
[04/22/23]seed@VM:~/.../09_hashing$ head -c 12320 pa > prefix
[04/22/23]seed@VM:~/.../09_hashing$ tail -c +12448 pa > suffix
[04/22/23]seed@VM:~/.../09_hashing$ md5collgen -p prefix -o prefix_and_P prefix_and_Q
MD5 collision generator v1.5
by Marc Stevens (http://www.win.tue.nl/hashclash/)

Using output filenames: 'prefix_and_P' and 'prefix_and_Q'
Using prefixfile: 'prefix'
Using initial value: fa3f7a62525b9c90471862a4a04139a5

Generating first block: ............................
Generating second block: S01..
Running time: 25.7183 s
[04/22/23]seed@VM:~/.../09_hashing$ cat prefix_and_P.suffix > program1.out
cat: prefix_and_P.suffix: No such file or directory
[04/22/23]seed@VM:~/.../09_hashing$ cat prefix_and_P suffix > program1.out
[04/22/23]seed@VM:~/.../09_hashing$ cat prefix_and_Q suffix > program1.out
[04/22/23]seed@VM:~/.../09_hashing$ diff program1.out program2.out
diff: program2.out: No such file or directory
[04/22/23]seed@VM:~/.../09_hashing$ cat prefix_and_Q suffix > program2.out
[04/22/23]seed@VM:~/.../09_hashing$ diff program1.out program2.out
[04/22/23]seed@VM:~/.../09_hashing$ md5sum program1.out
24811ae101a3609a474cf9db00acb790  program1.out
[04/22/23]seed@VM:~/.../09_hashing$ md5sum program2.out
24811ae101a3609a474cf9db00acb790  program2.out
[04/22/23]seed@VM:~/.../09_hashing$
```

There, its all complete and correct.