INFORMATION SECURITY

MD5 Collision Attack Lab

Ambreen Kanwal
University of Sahiwal

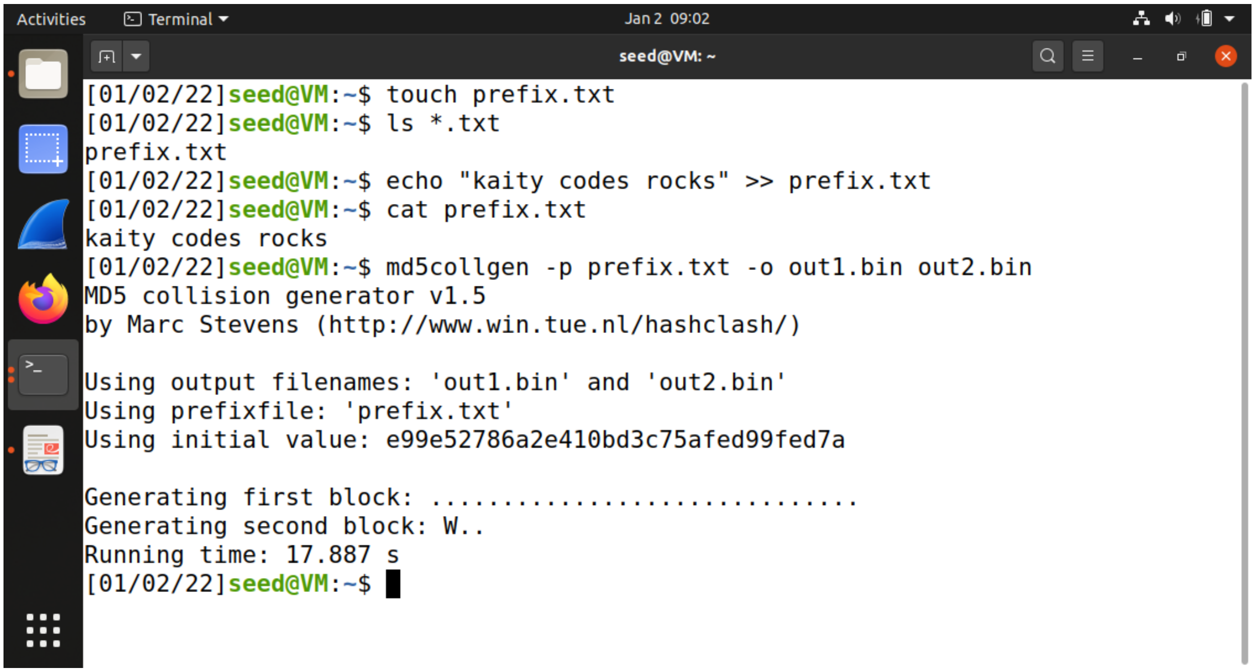
UNIVERSITY OF SAHIWAL

A Progressive University for Bright Future

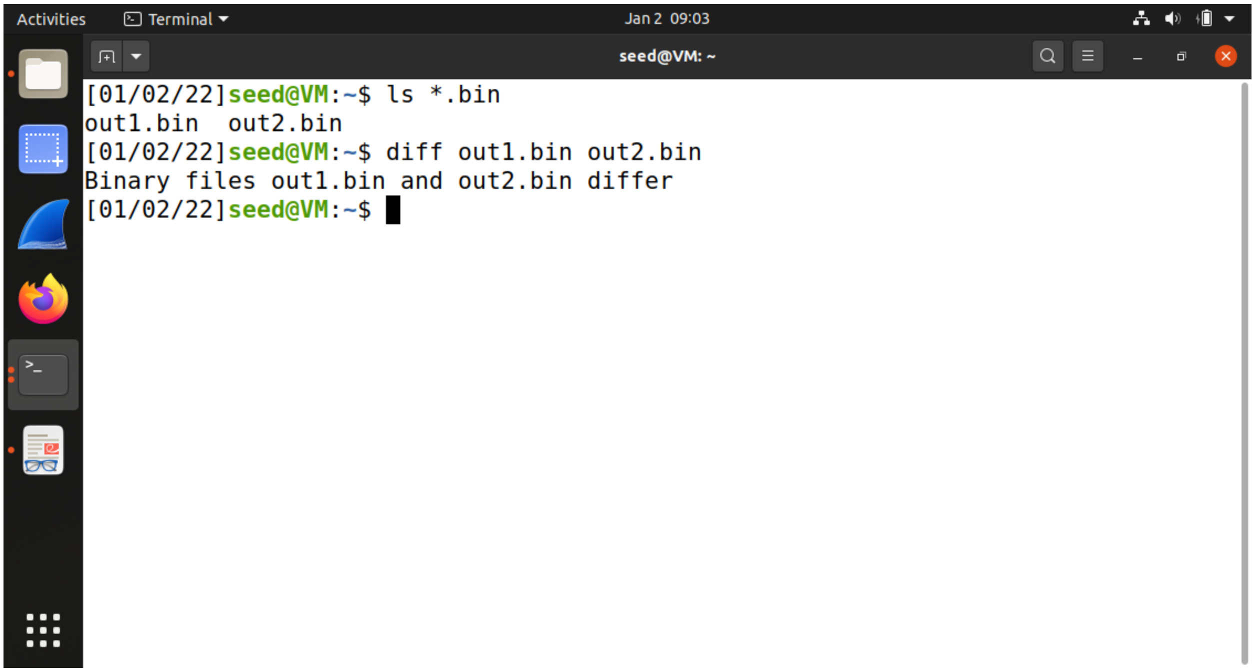
MD5 Collision Attack Lab

Task 1: Generating Two Different Files with the Same MD5 Hash

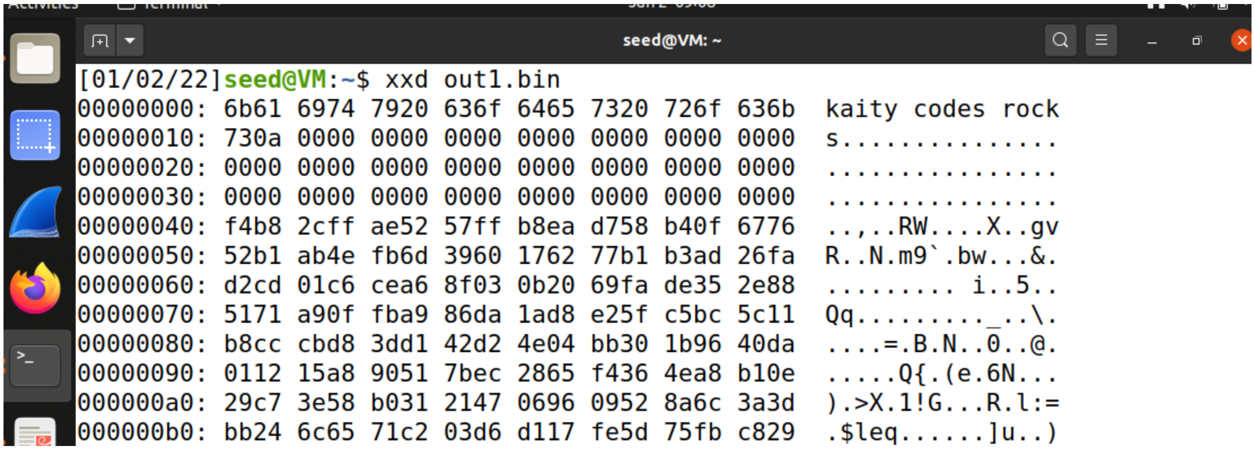
We will generate two different files with the same MD5 hash values. The beginning parts of these two files need to be the same. They share the same prefix. We can achieve this using the md5collgen program, which allows us to provide a prefix file with any arbitrary content.
The following command generates two output files, out1.bin and out2.bin, for a given a prefix file prefix.txt:
$ md5collgen -p prefix.txt -o out1.bin out2.bin



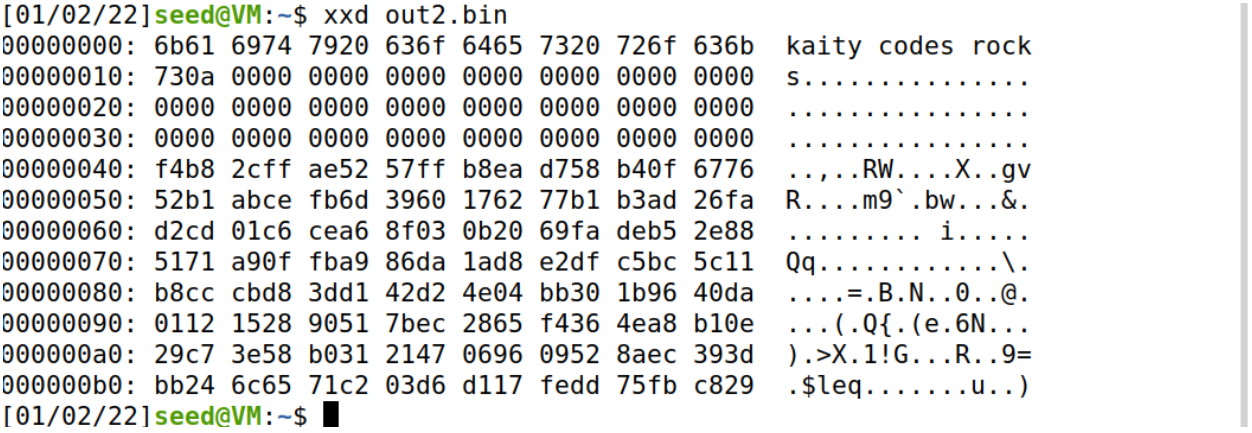
```
[01/02/22]seed@VM:~$ touch prefix.txt
[01/02/22]seed@VM:~$ ls *.txt
prefix.txt
[01/02/22]seed@VM:~$ echo "kaity codes rocks" >> prefix.txt
[01/02/22]seed@VM:~$ cat prefix.txt
kaity codes rocks
[01/02/22]seed@VM:~$ md5collgen -p prefix.txt -o out1.bin out2.bin
MD5 collision generator v1.5
by Marc Stevens (http://www.win.tue.nl/hashclash/)

Using output filenames: 'out1.bin' and 'out2.bin'
Using prefixfile: 'prefix.txt'
Using initial value: e99e52786a2e410bd3c75afed99fed7a

Generating first block: ..............................
Generating second block: W..
Running time: 17.887 s
[01/02/22]seed@VM:~$ 
```



```
[01/02/22]seed@VM:~$ ls *.bin
out1.bin  out2.bin
[01/02/22]seed@VM:~$ diff out1.bin out2.bin
Binary files out1.bin and out2.bin differ
[01/02/22]seed@VM:~$ 
```



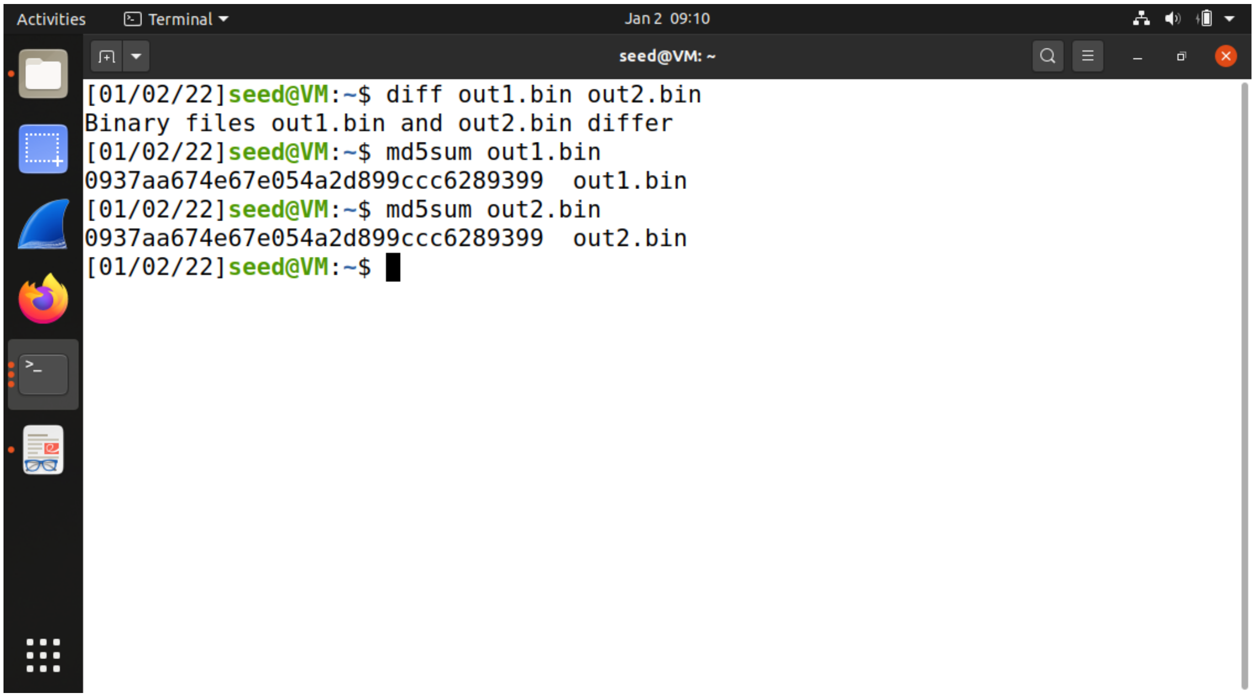
```
[01/02/22]seed@VM:~$ xxd out1.bin
00000000: 6b61 6974 7920 636f 6465 7320 726f 636b  kaity codes rock
00000010: 730a 0000 0000 0000 0000 0000 0000 0000  s...............
00000020: 0000 0000 0000 0000 0000 0000 0000 0000  ................
00000030: 0000 0000 0000 0000 0000 0000 0000 0000  ................
00000040: f4b8 2cff ae52 57ff b8ea d758 b40f 6776  ..,..RW....X..gv
00000050: 52b1 ab4e fb6d 3960 1762 77b1 b3ad 26fa  R..N.m9`.bw...&.
00000060: d2cd 01c6 cea6 8f03 0b20 69fa de35 2e88  ......... i..5..
00000070: 5171 a90f fba9 86da 1ad8 e25f c5bc 5c11  Qq.........._..\.
00000080: b8cc cbd8 3dd1 42d2 4e04 bb30 1b96 40da  ....=.B.N..0..@.
00000090: 0112 15a8 9051 7bec 2865 f436 4ea8 b10e  .....Q{.(e.6N...
000000a0: 29c7 3e58 b031 2147 0696 0952 8a6c 3a3d  ).>X.1!G...R.l:=
000000b0: bb24 6c65 71c2 03d6 d117 fe5d 75fb c829  .$leq......]u..)
```



```
[01/02/22]seed@VM:~$ xxd out2.bin
00000000: 6b61 6974 7920 636f 6465 7320 726f 636b  kaity codes rock
00000010: 730a 0000 0000 0000 0000 0000 0000 0000  s...............
00000020: 0000 0000 0000 0000 0000 0000 0000 0000  ................
00000030: 0000 0000 0000 0000 0000 0000 0000 0000  ................
00000040: f4b8 2cff ae52 57ff b8ea d758 b40f 6776  ..,..RW....X..gv
00000050: 52b1 abce fb6d 3960 1762 77b1 b3ad 26fa  R....m9`.bw...&.
00000060: d2cd 01c6 cea6 8f03 0b20 69fa deb5 2e88  ......... i.....
00000070: 5171 a90f fba9 86da 1ad8 e2df c5bc 5c11  Qq............\.
00000080: b8cc cbd8 3dd1 42d2 4e04 bb30 1b96 40da  ....=.B.N..0..@.
00000090: 0112 1528 9051 7bec 2865 f436 4ea8 b10e  ...(.Q{.(e.6N...
000000a0: 29c7 3e58 b031 2147 0696 0952 8aec 393d  ).>X.1!G...R..9=
000000b0: bb24 6c65 71c2 03d6 d117 fedd 75fb c829  .$leq.......u..)
[01/02/22]seed@VM:~$ 
```

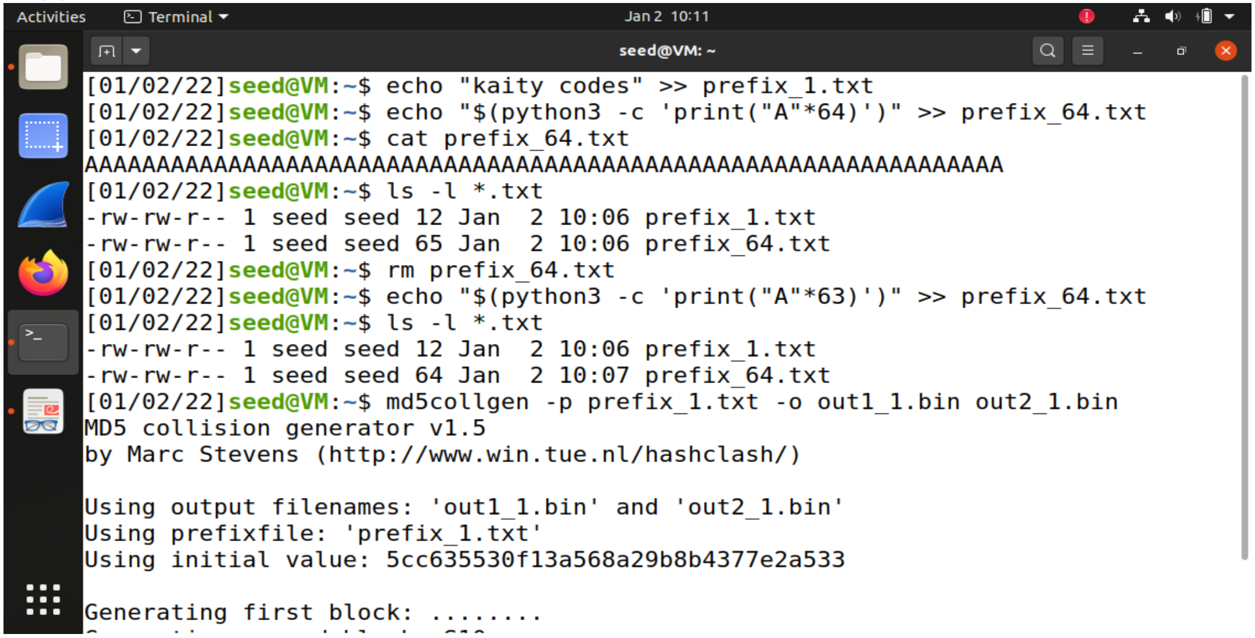
We can check whether the output files are distinct or not using the diff command. We can also use the md5sum command to check the MD5 hash of each output file. See the following commands.

$ diff out1.bin out2.bin
$ md5sum out1.bin
$ md5sum out2.bin

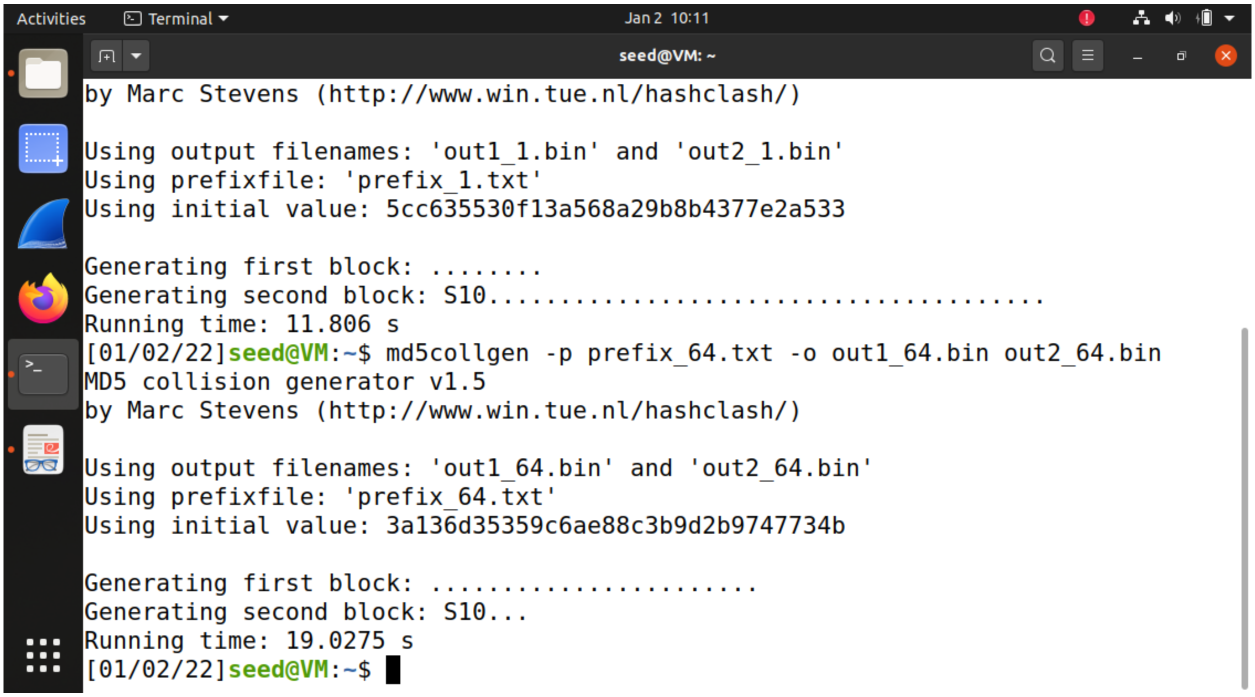


```
[01/02/22]seed@VM:~$ diff out1.bin out2.bin
Binary files out1.bin and out2.bin differ
[01/02/22]seed@VM:~$ md5sum out1.bin
0937aa674e67e054a2d899ccc6289399  out1.bin
[01/02/22]seed@VM:~$ md5sum out2.bin
0937aa674e67e054a2d899ccc6289399  out2.bin
[01/02/22]seed@VM:~$ 
```

– Question 1. If the length of your prefix file is not multiple of 64, what is going to happen?

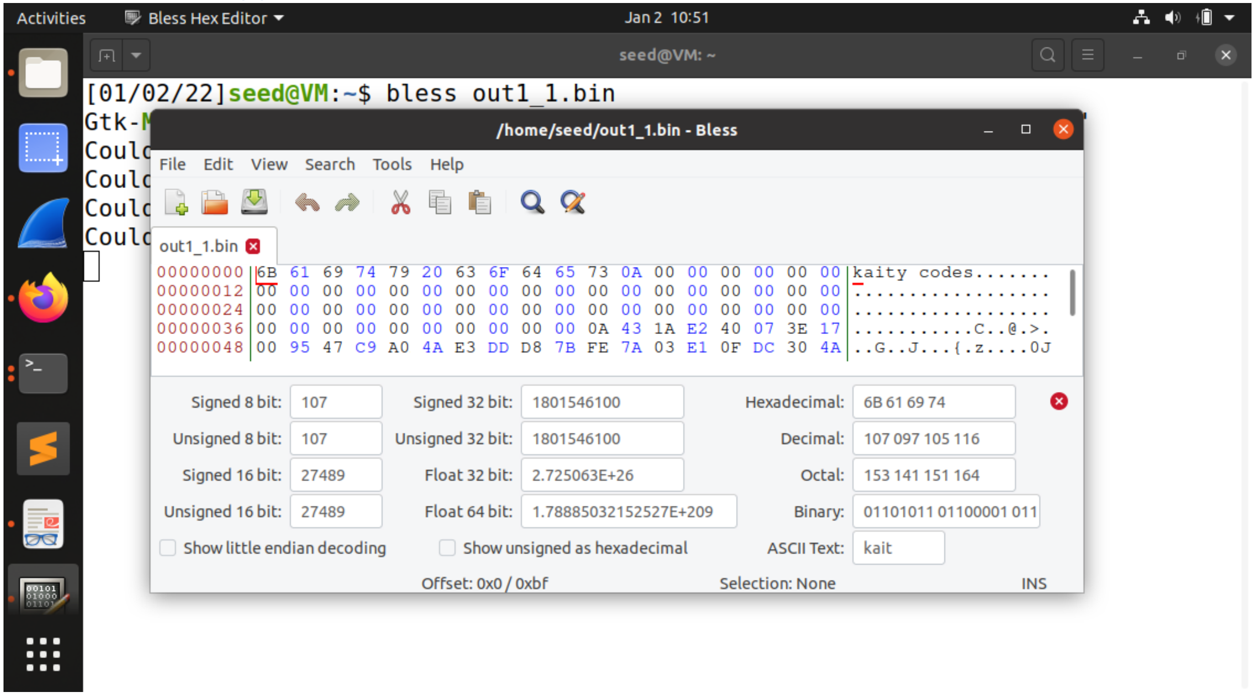


```
[01/02/22]seed@VM:~$ echo "kaity codes" >> prefix_1.txt
[01/02/22]seed@VM:~$ echo "$(python3 -c 'print("A"*64)')" >> prefix_64.txt
[01/02/22]seed@VM:~$ cat prefix_64.txt
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
[01/02/22]seed@VM:~$ ls -l *.txt
-rw-rw-r-- 1 seed seed 12 Jan  2 10:06 prefix_1.txt
-rw-rw-r-- 1 seed seed 65 Jan  2 10:06 prefix_64.txt
[01/02/22]seed@VM:~$ rm prefix_64.txt
[01/02/22]seed@VM:~$ echo "$(python3 -c 'print("A"*63)')" >> prefix_64.txt
[01/02/22]seed@VM:~$ ls -l *.txt
-rw-rw-r-- 1 seed seed 12 Jan  2 10:06 prefix_1.txt
-rw-rw-r-- 1 seed seed 64 Jan  2 10:07 prefix_64.txt
[01/02/22]seed@VM:~$ md5collgen -p prefix_1.txt -o out1_1.bin out2_1.bin
MD5 collision generator v1.5
by Marc Stevens (http://www.win.tue.nl/hashclash/)

Using output filenames: 'out1_1.bin' and 'out2_1.bin'
Using prefixfile: 'prefix_1.txt'
Using initial value: 5cc635530f13a568a29b8b4377e2a533

Generating first block: ........
```



```
by Marc Stevens (http://www.win.tue.nl/hashclash/)

Using output filenames: 'out1_1.bin' and 'out2_1.bin'
Using prefixfile: 'prefix_1.txt'
Using initial value: 5cc635530f13a568a29b8b4377e2a533

Generating first block: ........
Generating second block: S10.....................................
Running time: 11.806 s
[01/02/22]seed@VM:~$ md5collgen -p prefix_64.txt -o out1_64.bin out2_64.bin
MD5 collision generator v1.5
by Marc Stevens (http://www.win.tue.nl/hashclash/)

Using output filenames: 'out1_64.bin' and 'out2_64.bin'
Using prefixfile: 'prefix_64.txt'
Using initial value: 3a136d35359c6ae88c3b9d2b9747734b

Generating first block: .......................
Generating second block: S10...
Running time: 19.0275 s
[01/02/22]seed@VM:~$
```

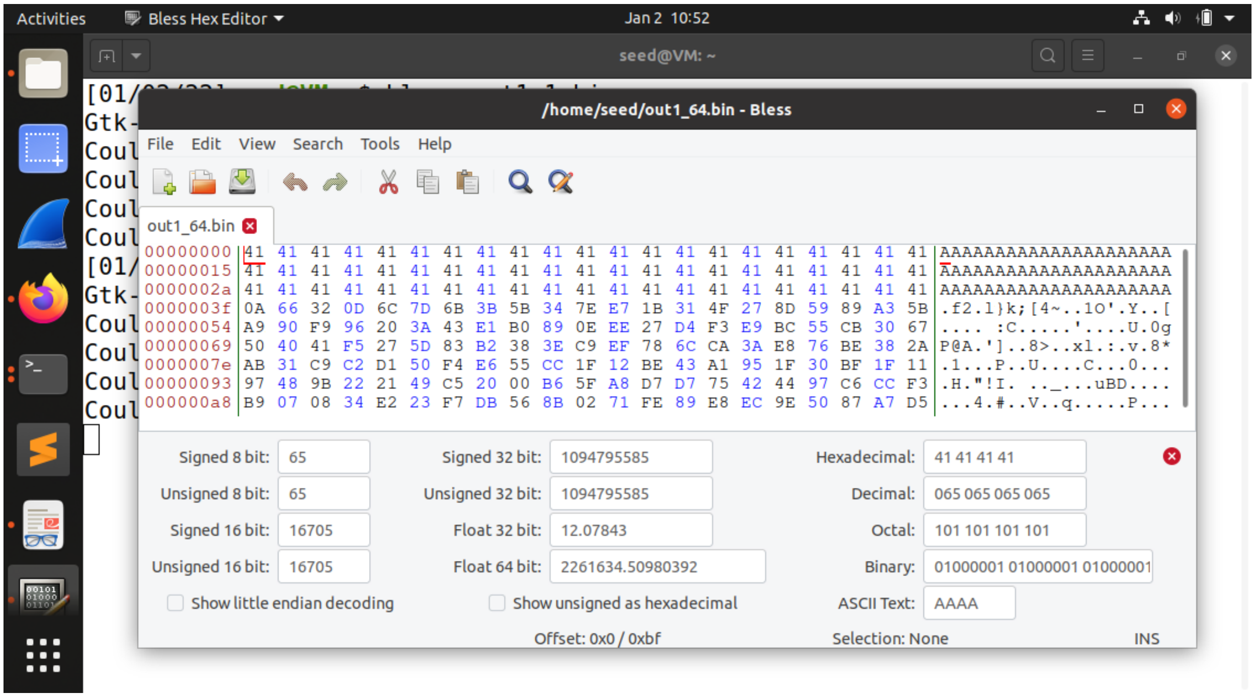
– Question 2. Create a prefix file with exactly 64 bytes, and run the collision tool again, and see what happens.



```
[01/02/22]seed@VM:~$ bless out1_1.bin
Gtk-M
Coulc
Coulc
Coulc
Coulc
```

```
00000000 6B 61 69 74 79 20 63 6F 64 65 73 0A 00 00 00 00 00 00 kaity codes.......
00000012 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ..................
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 0A 43 1A E2 40 07 3E 17 ...........C..@.>.
00000048 00 95 47 C9 A0 4A E3 DD D8 7B FE 7A 03 E1 0F DC 30 4A ..G..J...{.z....0J
```

Signed 8 bit: 107
Signed 32 bit: 1801546100
Hexadecimal: 6B 61 69 74
Unsigned 8 bit: 107
Unsigned 32 bit: 1801546100
Decimal: 107 097 105 116
Signed 16 bit: 27489
Float 32 bit: 2.725063E+26
Octal: 153 141 151 164
Unsigned 16 bit: 27489
Float 64 bit: 1.78885032152527E+209
Binary: 01101011 01100001 011
□ Show little endian decoding
□ Show unsigned as hexadecimal
ASCII Text: kait
Offset: 0x0 / 0xbf
Selection: None
INS

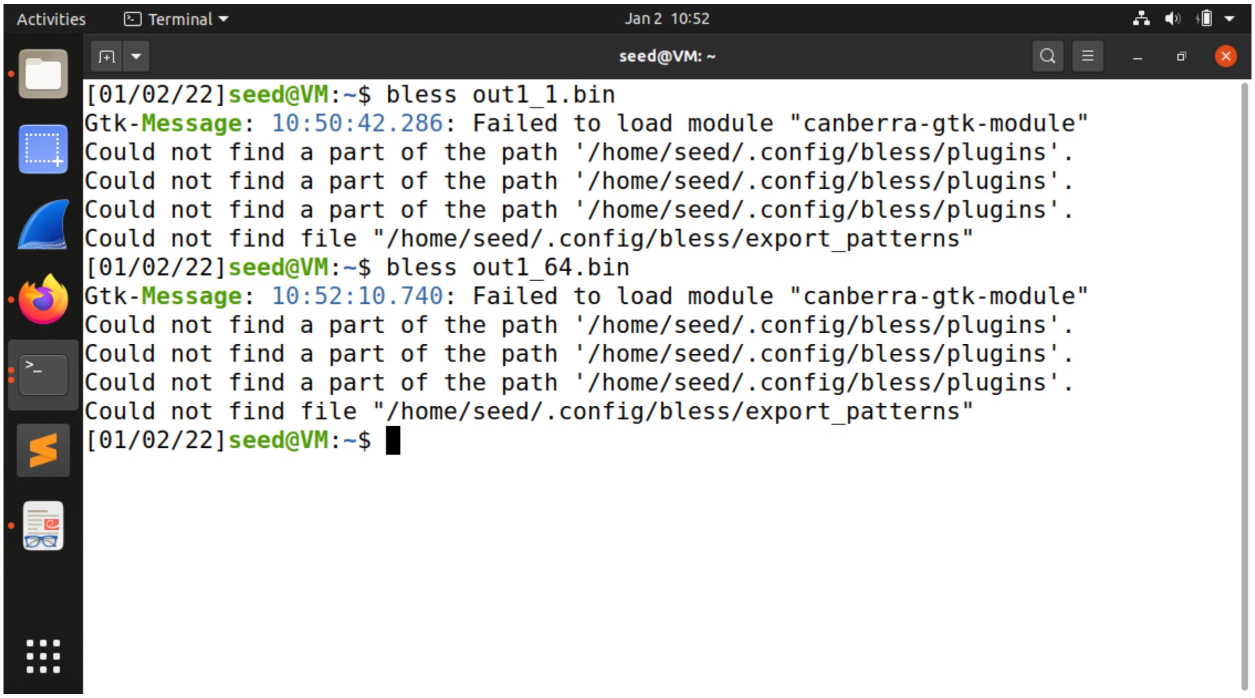

Activities
Bless Hex Editor
Jan 2 10:52
seed@VM: ~
/home/seed/out1_64.bin - Bless
File Edit View Search Tools Help
out1_64.bin

```
00000000 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 AAAAAAAAAAAAAAAAAAAAA
00000015 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 AAAAAAAAAAAAAAAAAAAAA
0000002a 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 AAAAAAAAAAAAAAAAAAAAA
0000003f 0A 66 32 0D 6C 7D 6B 3B 5B 34 7E E7 1B 31 4F 27 8D 59 89 A3 5B .f2.l}k;[4~..1O'.Y..[
00000054 A9 90 F9 96 20 3A 43 E1 B0 89 0E EE 27 D4 F3 E9 BC 55 CB 30 67 .... :C.....'....U.0g
00000069 50 40 41 F5 27 5D 83 B2 38 3E C9 EF 78 6C CA 3A E8 76 BE 38 2A P@A.']..8>..xl.:.v.8*
0000007e AB 31 C9 C2 D1 50 F4 E6 55 CC 1F 12 BE 43 A1 95 1F 30 BF 1F 11 .1...P..U....C...0...
00000093 97 48 9B 22 21 49 C5 20 00 B6 5F A8 D7 D7 75 42 44 97 C6 CC F3 .H."!I. .._...uBD....
000000a8 B9 07 08 34 E2 23 F7 DB 56 8B 02 71 FE 89 E8 EC 9E 50 87 A7 D5 ...4.#..V..q.....P...
```

Signed 8 bit: 65
Unsigned 8 bit: 65
Signed 16 bit: 16705
Unsigned 16 bit: 16705
Signed 32 bit: 1094795585
Unsigned 32 bit: 1094795585
Float 32 bit: 12.07843
Float 64 bit: 2261634.50980392
Hexadecimal: 41 41 41 41
Decimal: 065 065 065 065
Octal: 101 101 101 101
Binary: 01000001 01000001 01000001
ASCII Text: AAAA

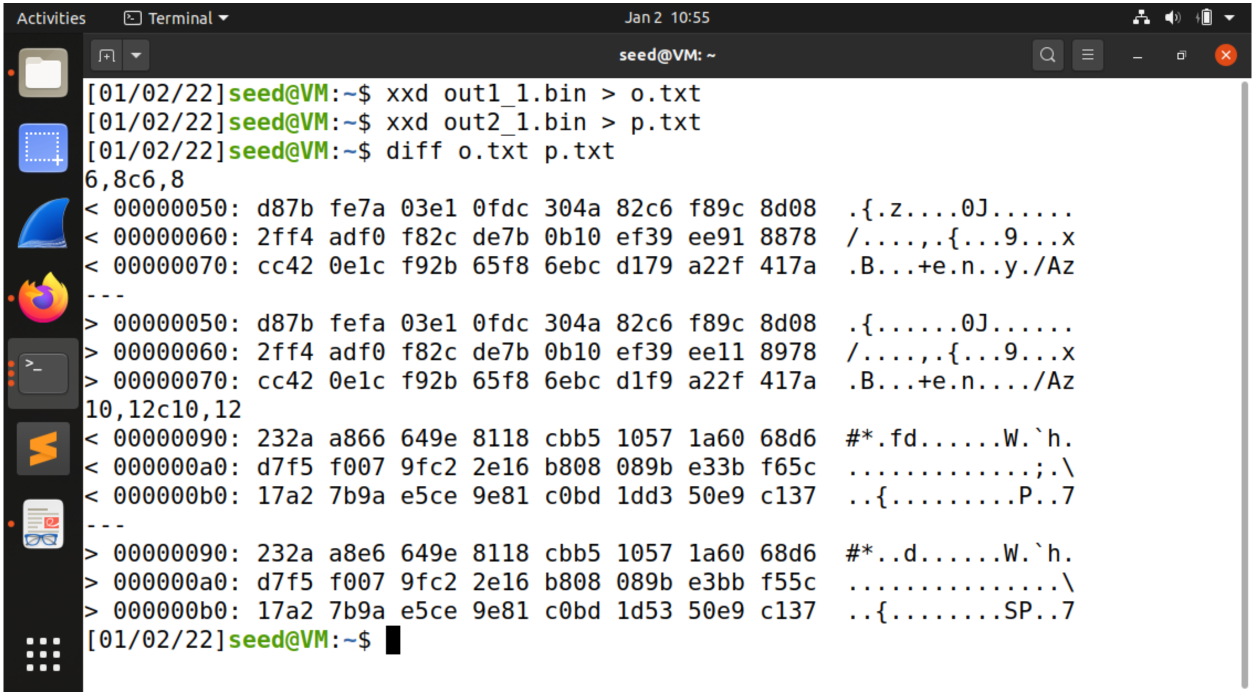
□ Show little endian decoding □ Show unsigned as hexadecimal

Offset: 0x0 / 0xbf Selection: None INS



```
[01/02/22]seed@VM:~$ bless out1_1.bin
Gtk-Message: 10:50:42.286: Failed to load module "canberra-gtk-module"
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"
[01/02/22]seed@VM:~$ bless out1_64.bin
Gtk-Message: 10:52:10.740: Failed to load module "canberra-gtk-module"
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"
[01/02/22]seed@VM:~$
```

– 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.



```
[01/02/22]seed@VM:~$ xxd out1_1.bin > o.txt
[01/02/22]seed@VM:~$ xxd out2_1.bin > p.txt
[01/02/22]seed@VM:~$ diff o.txt p.txt
6,8c6,8
< 00000050: d87b fe7a 03e1 0fdc 304a 82c6 f89c 8d08  .{.z....0J......
< 00000060: 2ff4 adf0 f82c de7b 0b10 ef39 ee91 8878  /....,.{...9...x
< 00000070: cc42 0e1c f92b 65f8 6ebc d179 a22f 417a  .B...+e.n..y./Az
---
> 00000050: d87b fefa 03e1 0fdc 304a 82c6 f89c 8d08  .{......0J......
> 00000060: 2ff4 adf0 f82c de7b 0b10 ef39 ee11 8978  /....,.{...9...x
> 00000070: cc42 0e1c f92b 65f8 6ebc d1f9 a22f 417a  .B...+e.n..../Az
10,12c10,12
< 00000090: 232a a866 649e 8118 cbb5 1057 1a60 68d6  #*.fd......W.`h.
< 000000a0: d7f5 f007 9fc2 2e16 b808 089b e33b f65c  .............;.\
< 000000b0: 17a2 7b9a e5ce 9e81 c0bd 1dd3 50e9 c137  ..{.........P..7
---
> 00000090: 232a a8e6 649e 8118 cbb5 1057 1a60 68d6  #*..d......W.`h.
> 000000a0: d7f5 f007 9fc2 2e16 b808 089b e3bb f55c  ...............\
> 000000b0: 17a2 7b9a e5ce 9e81 c0bd 1d53 50e9 c137  ..{........SP..7
[01/02/22]seed@VM:~$ 
```

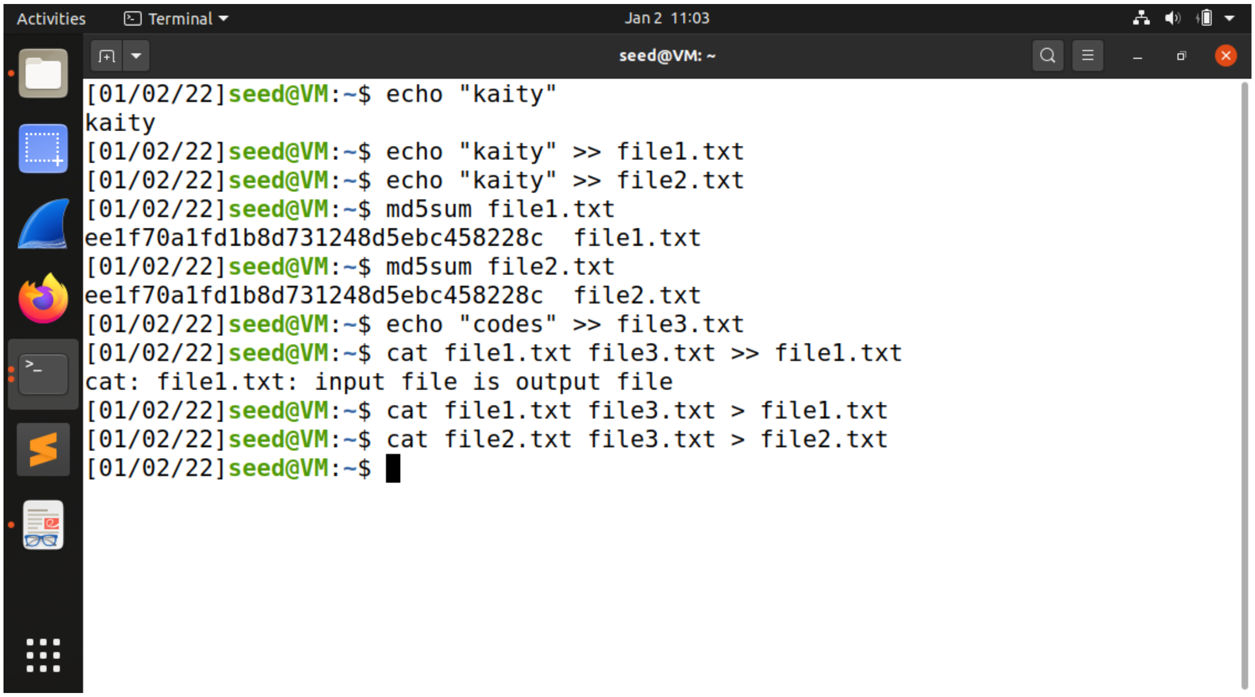
Task 2: Understanding MD5's Property

We will try to understand some of the properties of the MD5 algorithm. These properties are important for us to conduct further tasks in this lab. MD5 is a quite complicated algorithm, but from very high level, it is not so complicated.

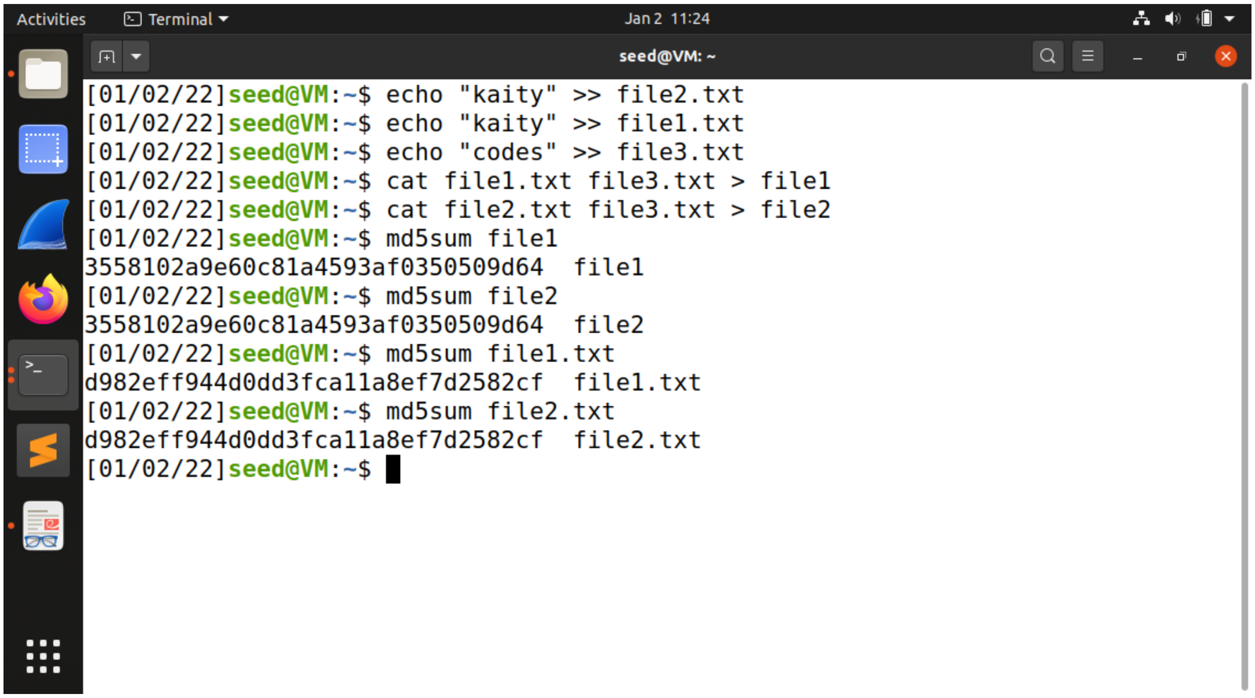
MD5 divides the input data into blocks of 64 bytes, and then computes the hash iteratively on these blocks. The core of the MD5 algorithm is a compression function, which takes two inputs, a 64-byte data block and the outcome of the previous iteration. The compression function produces a 128-bit IHV, which stands for "Intermediate Hash Value"; this output is then fed into the next iteration. If the current iteration is the last one, the IHV will be the final hash value. The IHV input for the first iteration (IHV0) is a fixed value.

Use the cat command to concatenate two files (binary or text files) into one. The following command concatenates the contents of file2 to the contents of file1, and places the result in file3.

$ cat file1 file2 > file3

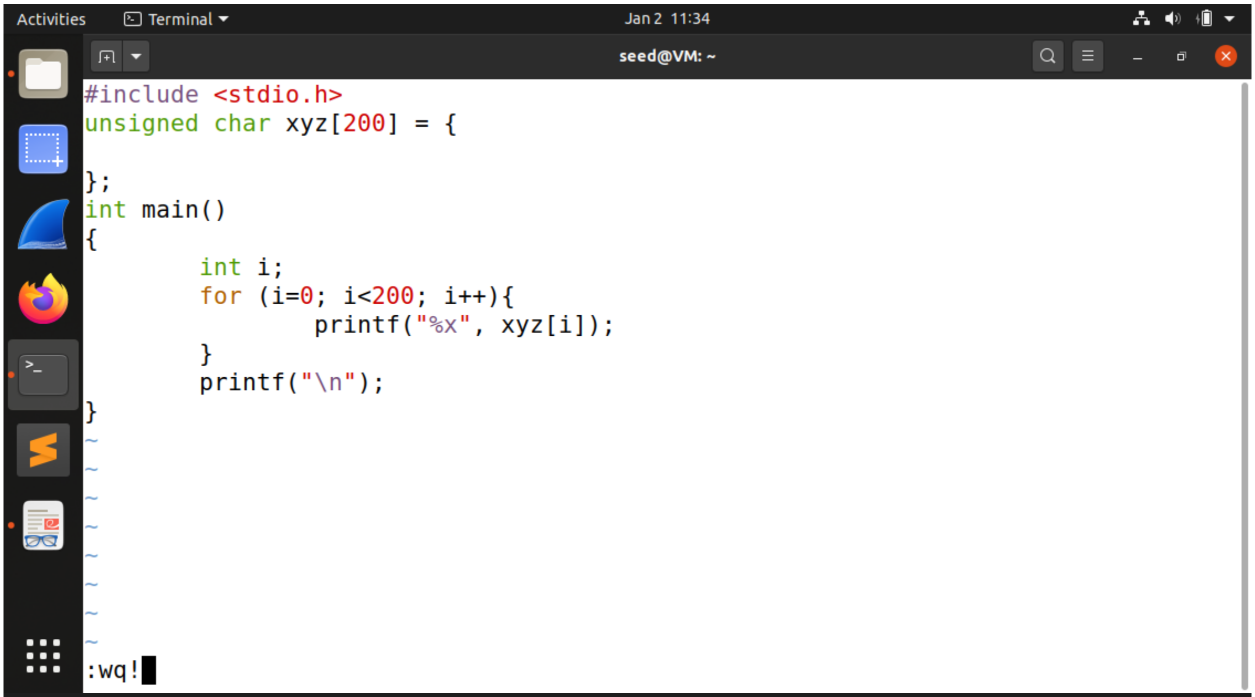


```
[01/02/22]seed@VM:~$ echo "kaity"
kaity
[01/02/22]seed@VM:~$ echo "kaity" >> file1.txt
[01/02/22]seed@VM:~$ echo "kaity" >> file2.txt
[01/02/22]seed@VM:~$ md5sum file1.txt
ee1f70a1fd1b8d731248d5ebc458228c  file1.txt
[01/02/22]seed@VM:~$ md5sum file2.txt
ee1f70a1fd1b8d731248d5ebc458228c  file2.txt
[01/02/22]seed@VM:~$ echo "codes" >> file3.txt
[01/02/22]seed@VM:~$ cat file1.txt file3.txt >> file1.txt
cat: file1.txt: input file is output file
[01/02/22]seed@VM:~$ cat file1.txt file3.txt > file1.txt
[01/02/22]seed@VM:~$ cat file2.txt file3.txt > file2.txt
[01/02/22]seed@VM:~$ 
```

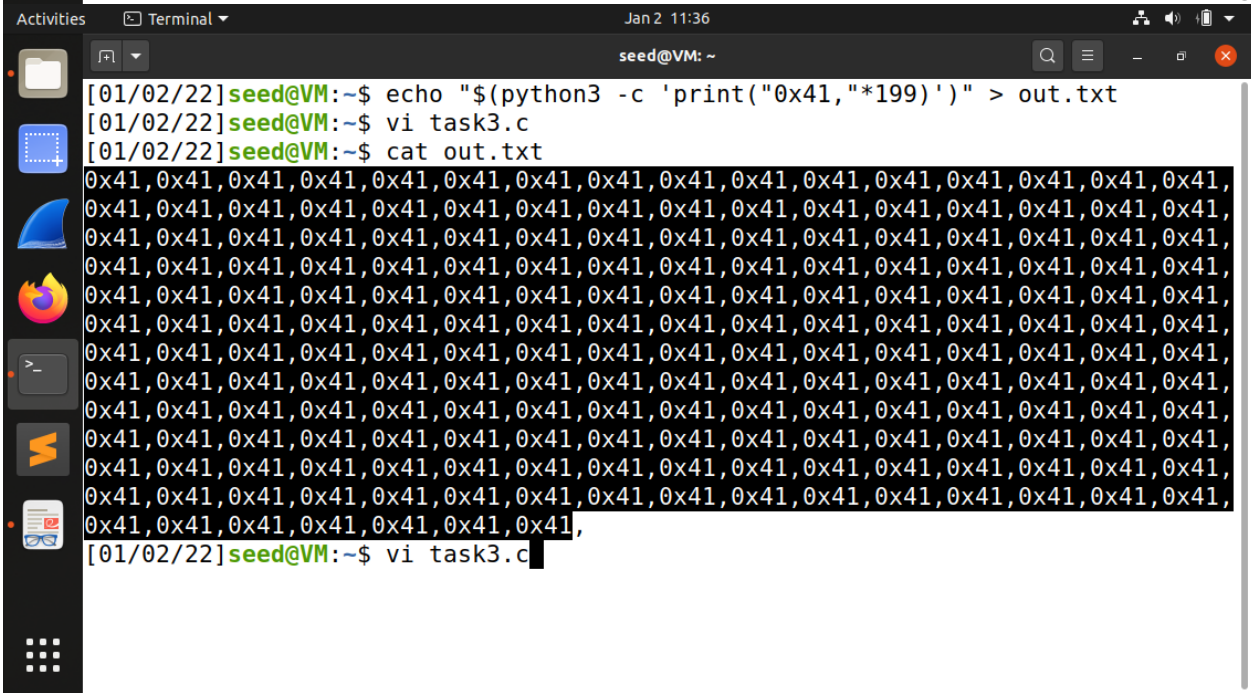


```
[01/02/22]seed@VM:~$ echo "kaity" >> file2.txt
[01/02/22]seed@VM:~$ echo "kaity" >> file1.txt
[01/02/22]seed@VM:~$ echo "codes" >> file3.txt
[01/02/22]seed@VM:~$ cat file1.txt file3.txt > file1
[01/02/22]seed@VM:~$ cat file2.txt file3.txt > file2
[01/02/22]seed@VM:~$ md5sum file1
3558102a9e60c81a4593af0350509d64  file1
[01/02/22]seed@VM:~$ md5sum file2
3558102a9e60c81a4593af0350509d64  file2
[01/02/22]seed@VM:~$ md5sum file1.txt
d982eff944d0dd3fca11a8ef7d2582cf  file1.txt
[01/02/22]seed@VM:~$ md5sum file2.txt
d982eff944d0dd3fca11a8ef7d2582cf  file2.txt
[01/02/22]seed@VM:~$ 
```

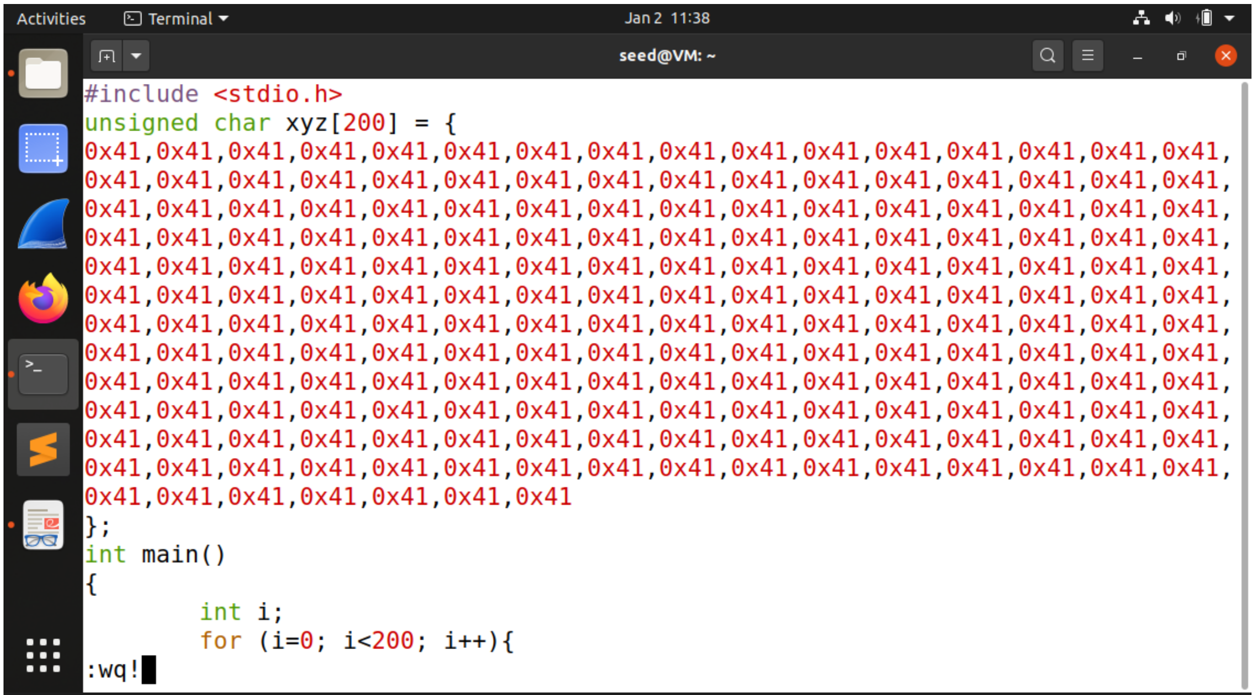
Task 3: Generating Two Executable Files with the Same MD5 Hash



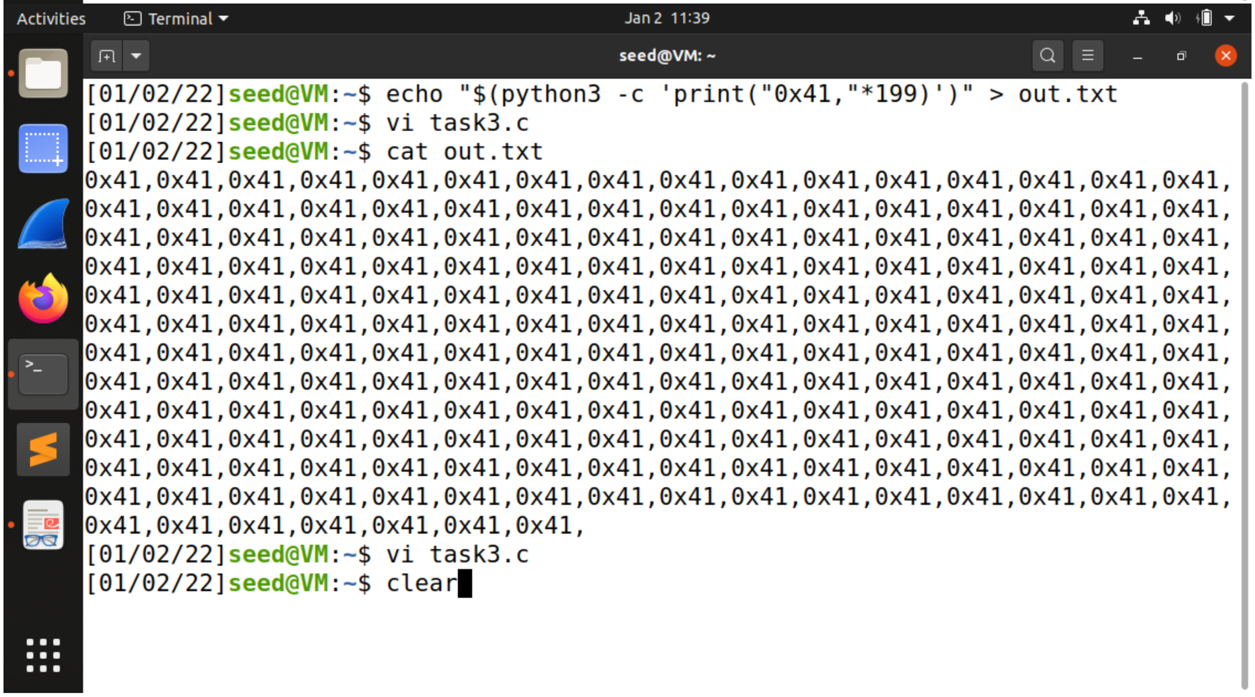
```
#include <stdio.h>
unsigned char xyz[200] = {

};
int main()
{
        int i;
        for (i=0; i<200; i++){
                printf("%x", xyz[i]);
        }
        printf("\n");
}
~
~
~
~
~
~
~
~
:wq!
```



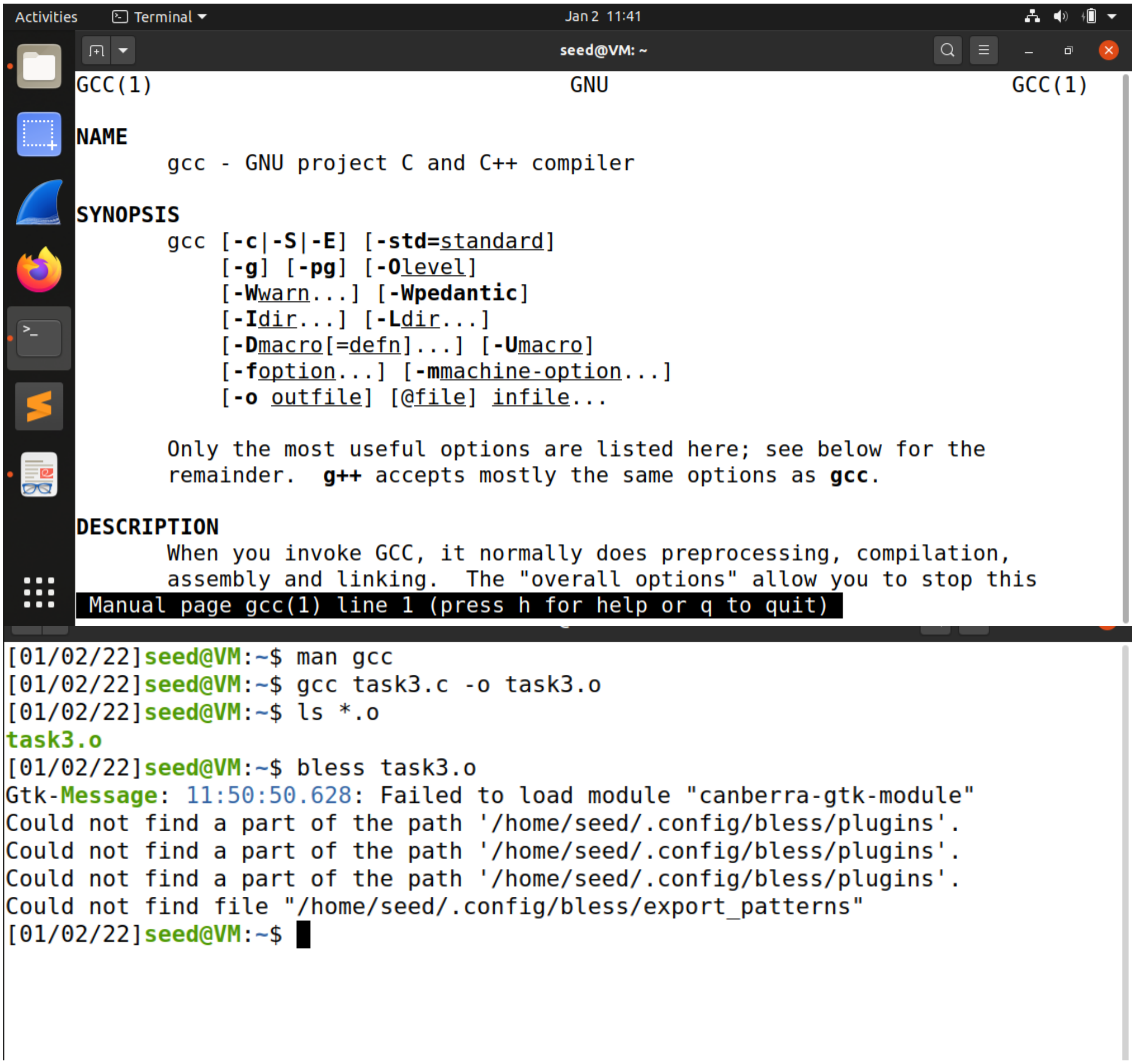
```
[01/02/22]seed@VM:~$ echo "$(python3 -c 'print("0x41,"*199)')" > out.txt
[01/02/22]seed@VM:~$ vi task3.c
[01/02/22]seed@VM:~$ cat out.txt
0x41,0x41,0x41,0x41,0x41,0x41,0x41,0x41,0x41,0x41,0x41,0x41,0x41,0x41,0x41,0x41,
0x41,0x41,0x41,0x41,0x41,0x41,0x41,0x41,0x41,0x41,0x41,0x41,0x41,0x41,0x41,0x41,
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,
[01/02/22]seed@VM:~$ vi task3.c
```



```
#include <stdio.h>
unsigned char xyz[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
};
int main()
{
        int i;
        for (i=0; i<200; i++){
:wq!
```

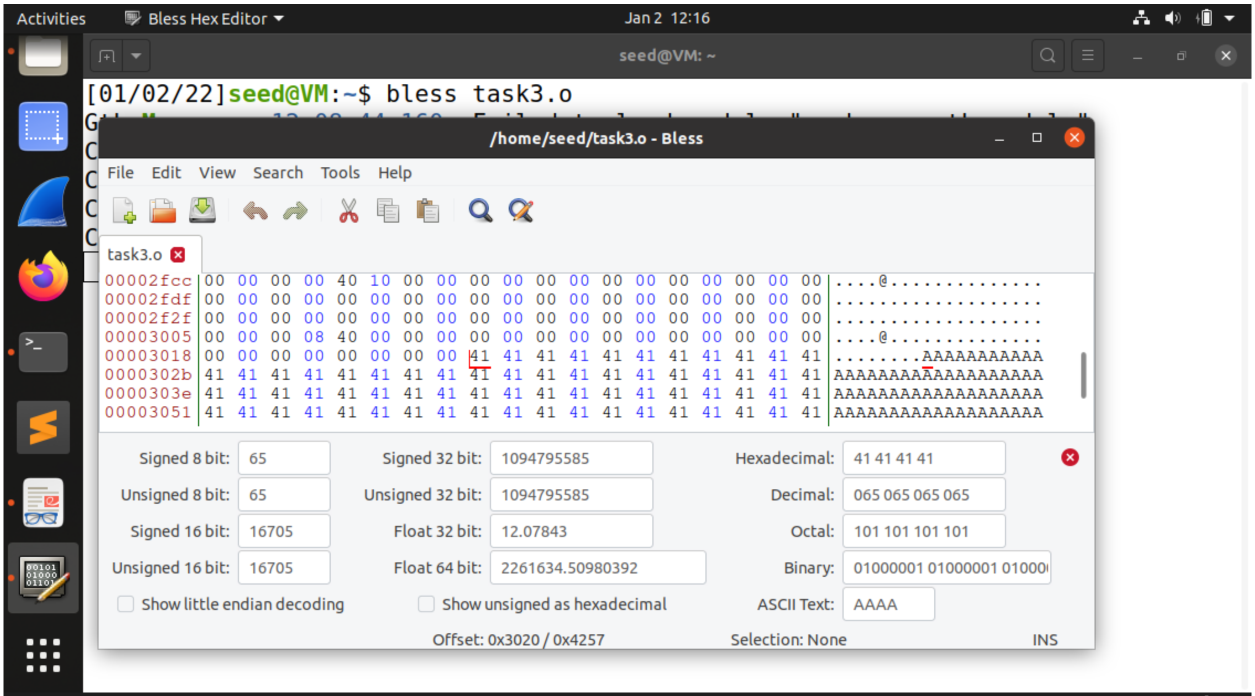


```
[01/02/22]seed@VM:~$ echo "$(python3 -c 'print("0x41,"*199)')" > out.txt
[01/02/22]seed@VM:~$ vi task3.c
[01/02/22]seed@VM:~$ cat out.txt
0x41,0x41,0x41,0x41,0x41,0x41,0x41,0x41,0x41,0x41,0x41,0x41,0x41,0x41,0x41,0x41,
0x41,0x41,0x41,0x41,0x41,0x41,0x41,0x41,0x41,0x41,0x41,0x41,0x41,0x41,0x41,0x41,
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,
[01/02/22]seed@VM:~$ vi task3.c
[01/02/22]seed@VM:~$ clear
```



```
GCC(1)                              GNU                              GCC(1)

NAME
       gcc - GNU project C and C++ compiler

SYNOPSIS
       gcc [-c|-S|-E] [-std=standard]
           [-g] [-pg] [-Olevel]
           [-Wwarn...] [-Wpedantic]
           [-Idir...] [-Ldir...]
           [-Dmacro[=defn]...] [-Umacro]
           [-foption...] [-mmachine-option...]
           [-o outfile] [@file] infile...

       Only the most useful options are listed here; see below for the
       remainder.  g++ accepts mostly the same options as gcc.

DESCRIPTION
       When you invoke GCC, it normally does preprocessing, compilation,
       assembly and linking.  The "overall options" allow you to stop this
 Manual page gcc(1) line 1 (press h for help or q to quit)
```

```
[01/02/22]seed@VM:~$ man gcc
[01/02/22]seed@VM:~$ gcc task3.c -o task3.o
[01/02/22]seed@VM:~$ ls *.o
task3.o
[01/02/22]seed@VM:~$ bless task3.o
Gtk-Message: 11:50:50.628: Failed to load module "canberra-gtk-module"
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"
[01/02/22]seed@VM:~$ 
```

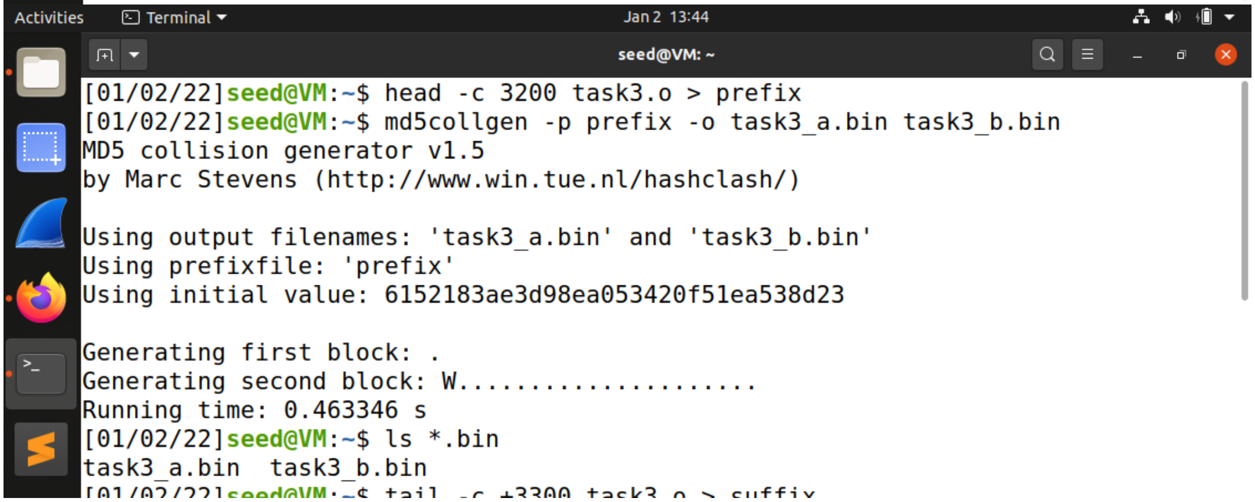


```
[01/02/22]seed@VM:~$ bless task3.o
```

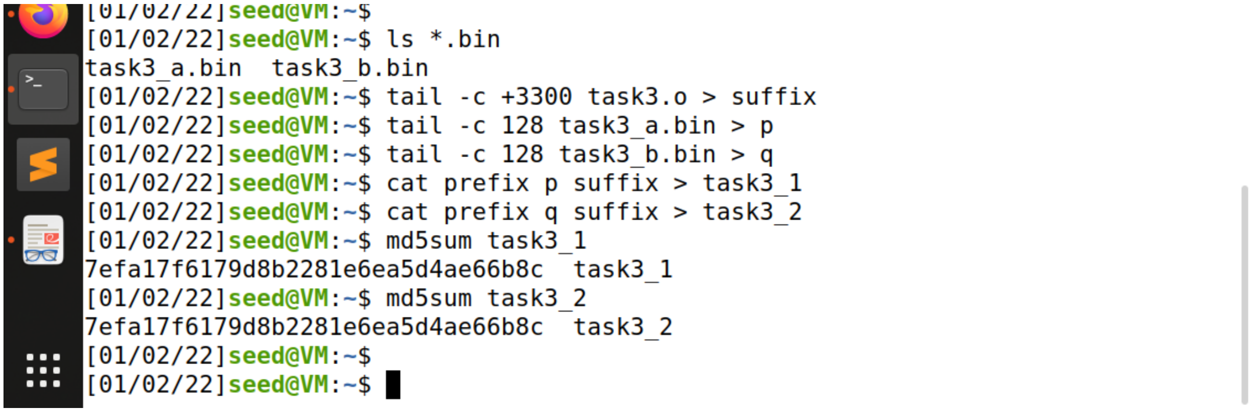
```
00002fcc 00 00 00 00 40 10 00 00 00 00 00 00 00 00 00 00 00 00 00 ....@..............
00002fdf 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ...................
00002f2f 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ...................
00003005 00 00 00 08 40 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ....@..............
00003018 00 00 00 00 00 00 00 00 41 41 41 41 41 41 41 41 41 41 41 ........AAAAAAAAAAA
0000302b 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 AAAAAAAAAAAAAAAAAAA
0000303e 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 AAAAAAAAAAAAAAAAAAA
00003051 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 AAAAAAAAAAAAAAAAAAA
```

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 01000
ASCII Text: AAAA

Offset: 0x3020 / 0x4257 Selection: None INS

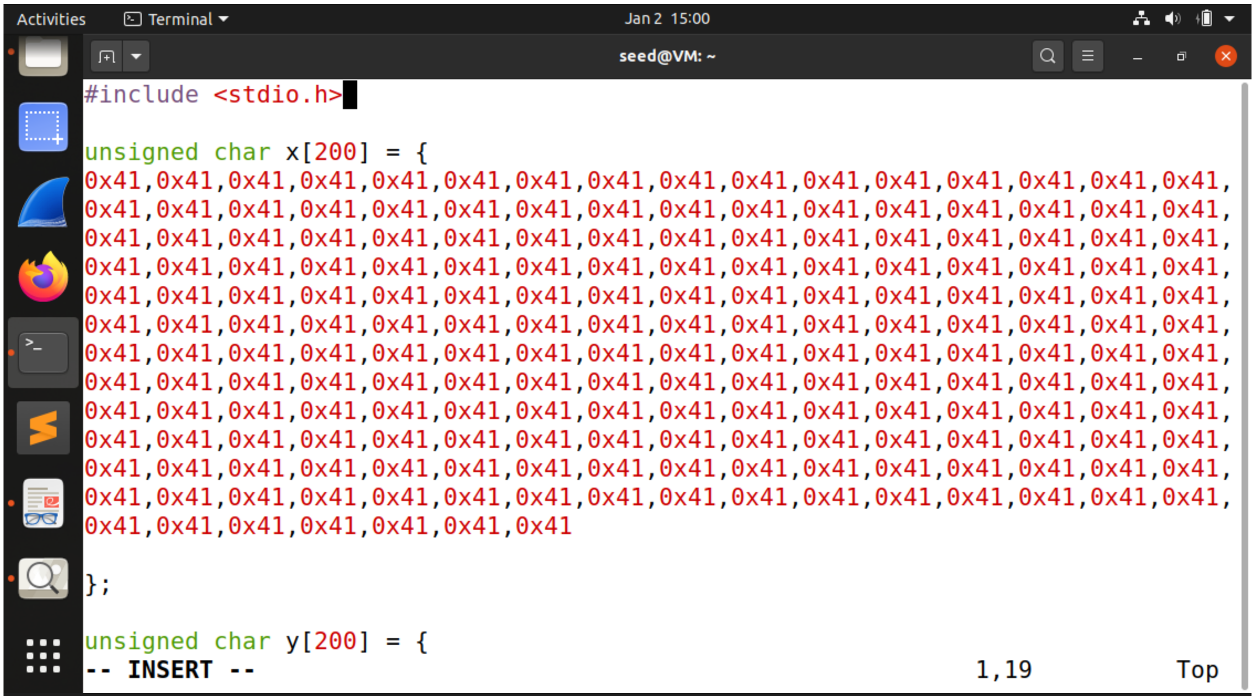


```
[01/02/22]seed@VM:~$ head -c 3200 task3.o > prefix
[01/02/22]seed@VM:~$ md5collgen -p prefix -o task3_a.bin task3_b.bin
MD5 collision generator v1.5
by Marc Stevens (http://www.win.tue.nl/hashclash/)

Using output filenames: 'task3_a.bin' and 'task3_b.bin'
Using prefixfile: 'prefix'
Using initial value: 6152183ae3d98ea053420f51ea538d23

Generating first block: .
Generating second block: W....................
Running time: 0.463346 s
[01/02/22]seed@VM:~$ ls *.bin
task3_a.bin  task3_b.bin
[01/02/22]seed@VM:~$ tail -c +3300 task3.o > suffix
```

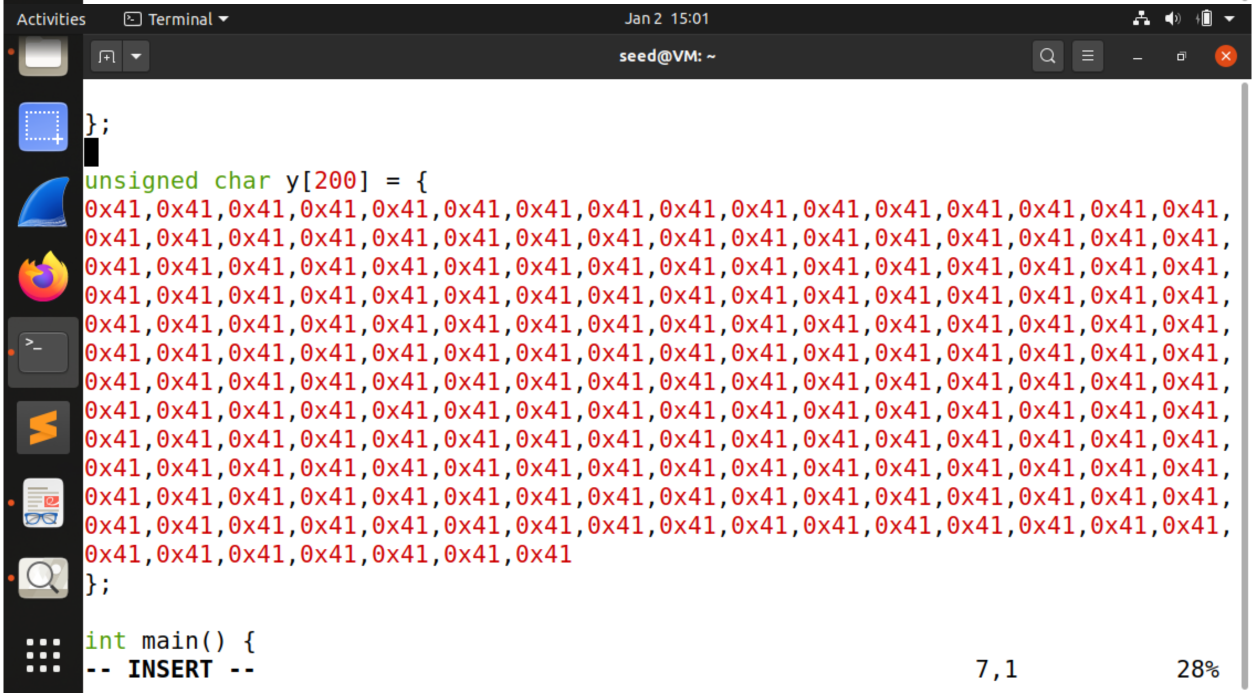


```
[01/02/22]seed@VM:~$
[01/02/22]seed@VM:~$ ls *.bin
task3_a.bin  task3_b.bin
[01/02/22]seed@VM:~$ tail -c +3300 task3.o > suffix
[01/02/22]seed@VM:~$ tail -c 128 task3_a.bin > p
[01/02/22]seed@VM:~$ tail -c 128 task3_b.bin > q
[01/02/22]seed@VM:~$ cat prefix p suffix > task3_1
[01/02/22]seed@VM:~$ cat prefix q suffix > task3_2
[01/02/22]seed@VM:~$ md5sum task3_1
7efa17f6179d8b2281e6ea5d4ae66b8c  task3_1
[01/02/22]seed@VM:~$ md5sum task3_2
7efa17f6179d8b2281e6ea5d4ae66b8c  task3_2
[01/02/22]seed@VM:~$
[01/02/22]seed@VM:~$
```

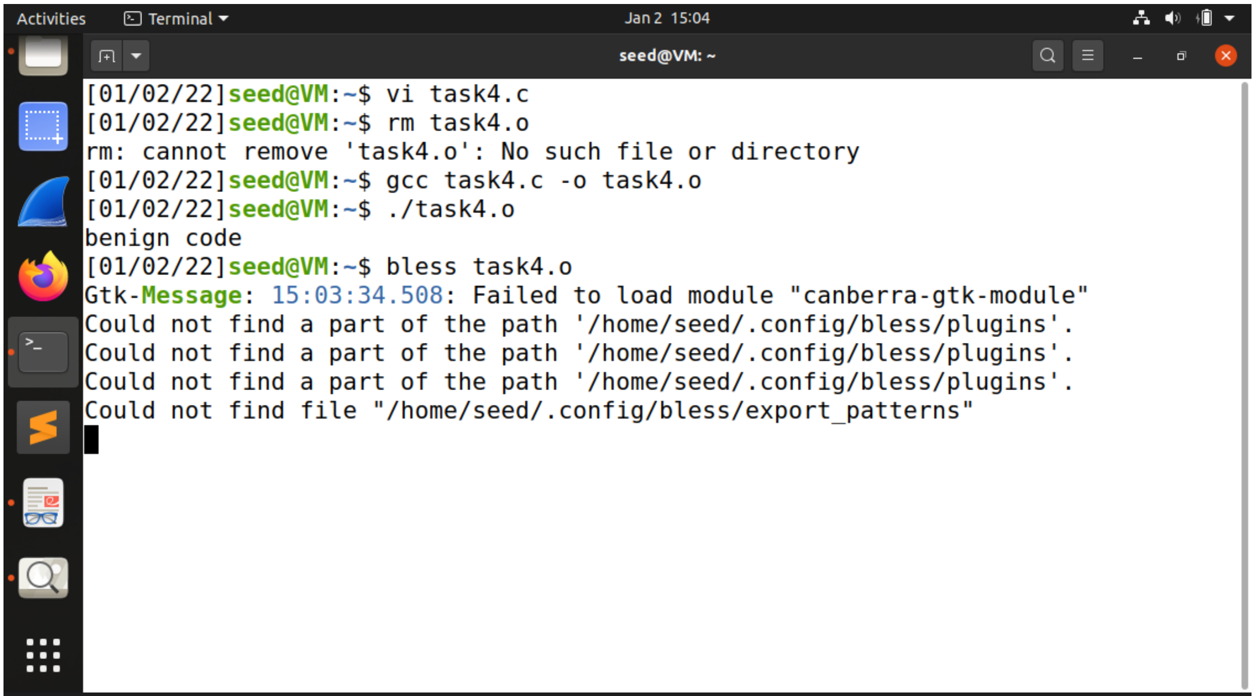
Task 4: Making the Two Programs Behave Differently



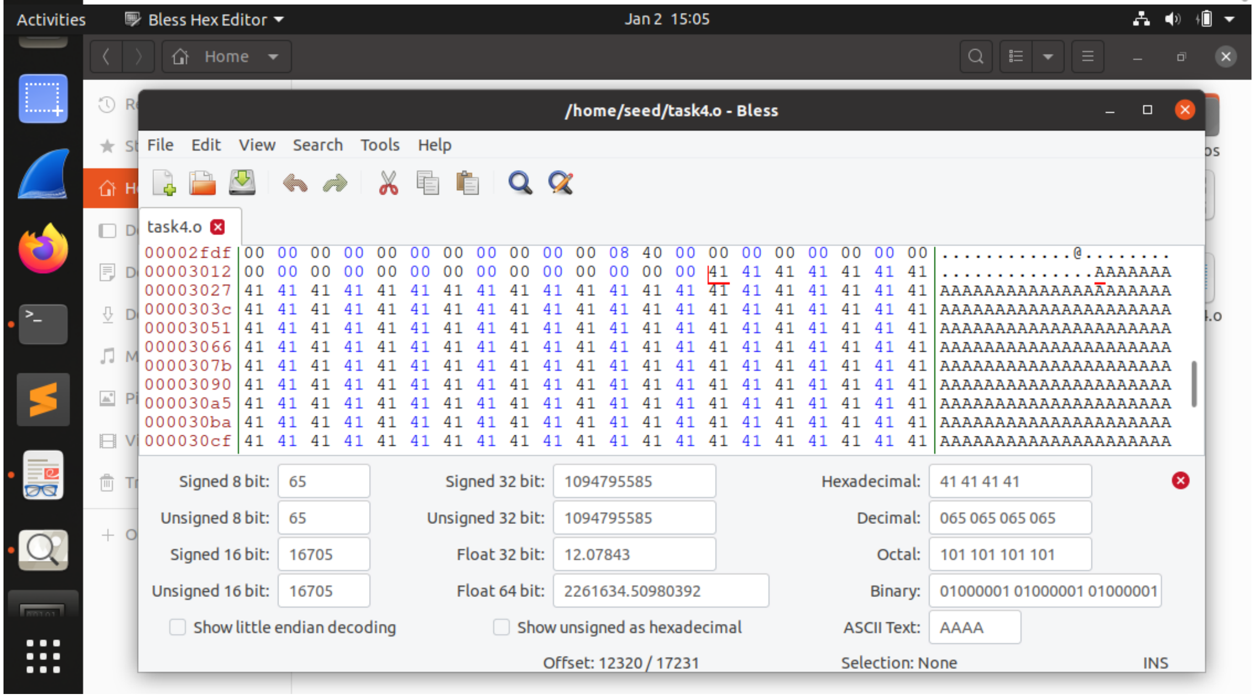
```
#include <stdio.h>

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

};

unsigned char y[200] = {
-- INSERT --                                                   1,19          Top
```



```

};

unsigned char y[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
};

int main() {
-- INSERT --                                                   7,1           28%
```



```
[01/02/22]seed@VM:~$ vi task4.c
[01/02/22]seed@VM:~$ rm task4.o
rm: cannot remove 'task4.o': No such file or directory
[01/02/22]seed@VM:~$ gcc task4.c -o task4.o
[01/02/22]seed@VM:~$ ./task4.o
benign code
[01/02/22]seed@VM:~$ bless task4.o
Gtk-Message: 15:03:34.508: Failed to load module "canberra-gtk-module"
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"
```

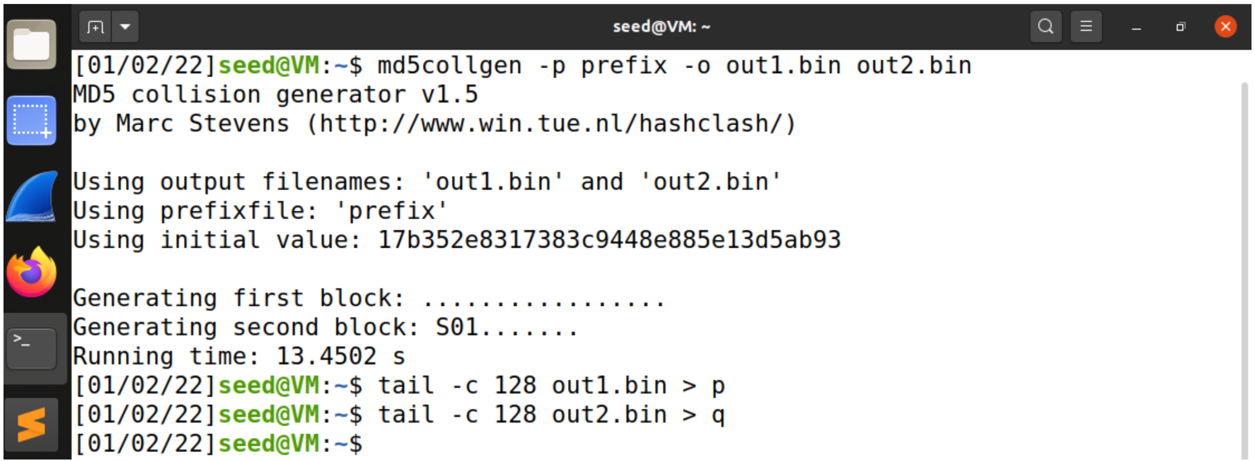


```
00002fdf 00 00 00 00 00 00 00 00 00 00 00 08 40 00 00 00 00 00 00 00 00 ............@........
00003012 00 00 00 00 00 00 00 00 00 00 00 00 00 00 41 41 41 41 41 41 41 ..............AAAAAAA
00003027 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 AAAAAAAAAAAAAAAAAAAAA
0000303c 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 AAAAAAAAAAAAAAAAAAAAA
00003051 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 AAAAAAAAAAAAAAAAAAAAA
00003066 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 AAAAAAAAAAAAAAAAAAAAA
0000307b 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 AAAAAAAAAAAAAAAAAAAAA
00003090 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 AAAAAAAAAAAAAAAAAAAAA
000030a5 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 AAAAAAAAAAAAAAAAAAAAA
000030ba 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 AAAAAAAAAAAAAAAAAAAAA
000030cf 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 AAAAAAAAAAAAAAAAAAAAA
```

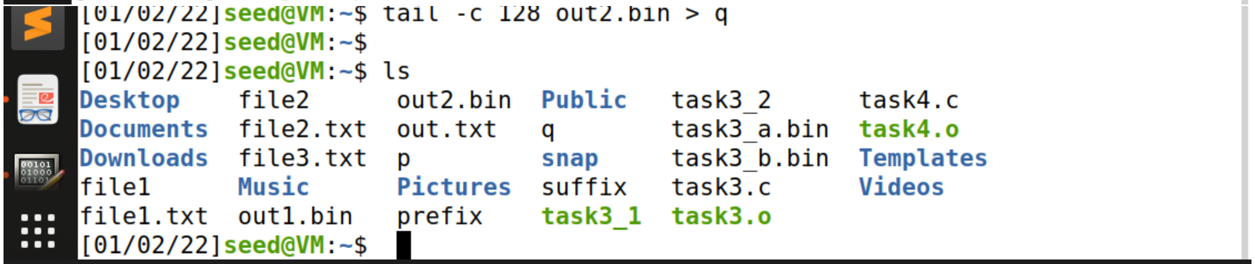
Signed 8 bit: 65
Unsigned 8 bit: 65
Signed 16 bit: 16705
Unsigned 16 bit: 16705
Signed 32 bit: 1094795585
Unsigned 32 bit: 1094795585
Float 32 bit: 12.07843
Float 64 bit: 2261634.50980392
Hexadecimal: 41 41 41 41
Decimal: 065 065 065 065
Octal: 101 101 101 101
Binary: 01000001 01000001 01000001
ASCII Text: AAAA

□ Show little endian decoding
□ Show unsigned as hexadecimal

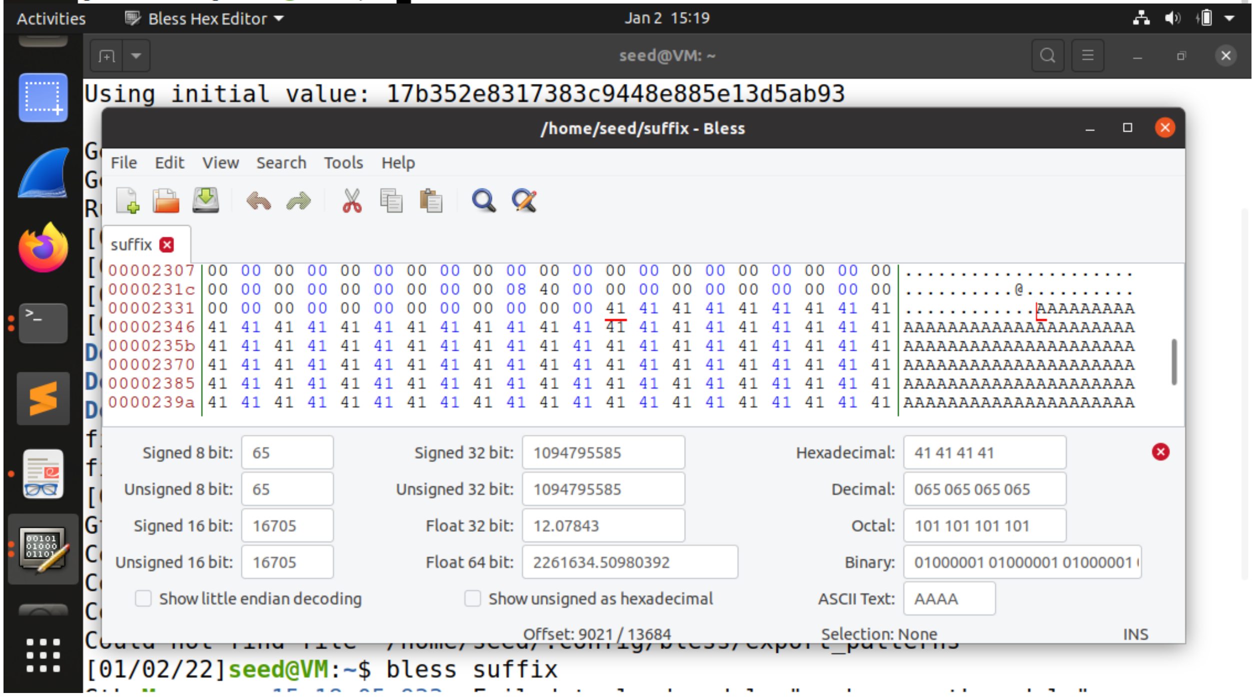
Offset: 12320 / 17231 Selection: None INS



```
[01/02/22]seed@VM:~$ md5collgen -p prefix -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'
Using initial value: 17b352e8317383c9448e885e13d5ab93

Generating first block: .................
Generating second block: S01.......
Running time: 13.4502 s
[01/02/22]seed@VM:~$ tail -c 128 out1.bin > p
[01/02/22]seed@VM:~$ tail -c 128 out2.bin > q
[01/02/22]seed@VM:~$
```



```
[01/02/22]seed@VM:~$ tail -c 128 out2.bin > q
[01/02/22]seed@VM:~$
[01/02/22]seed@VM:~$ ls
Desktop     file2      out2.bin  Public   task3_2      task4.c
Documents   file2.txt  out.txt   q        task3_a.bin  task4.o
Downloads   file3.txt  p         snap     task3_b.bin  Templates
file1       Music      Pictures  suffix   task3.c      Videos
file1.txt   out1.bin   prefix    task3_1  task3.o
[01/02/22]seed@VM:~$ 
```



```
Using initial value: 17b352e8317383c9448e885e13d5ab93
```

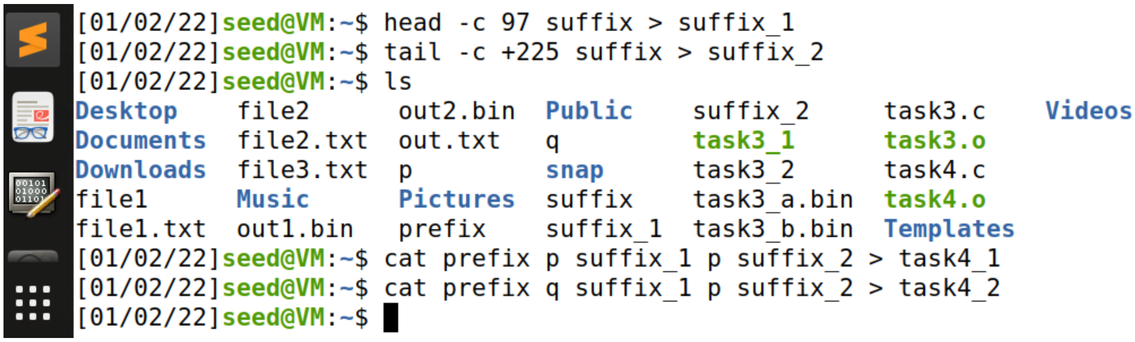
```
00002307 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 .....................
0000231c 00 00 00 00 00 00 00 00 00 08 40 00 00 00 00 00 00 00 00 00 00 ..........@..........
00002331 00 00 00 00 00 00 00 00 00 00 00 00 41 41 41 41 41 41 41 41 41 ............AAAAAAAAA
00002346 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 AAAAAAAAAAAAAAAAAAAAA
0000235b 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 AAAAAAAAAAAAAAAAAAAAA
00002370 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 AAAAAAAAAAAAAAAAAAAAA
00002385 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 AAAAAAAAAAAAAAAAAAAAA
0000239a 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 AAAAAAAAAAAAAAAAAAAAA
```

Signed 8 bit: 65
Unsigned 8 bit: 65
Signed 16 bit: 16705
Unsigned 16 bit: 16705
Signed 32 bit: 1094795585
Unsigned 32 bit: 1094795585
Float 32 bit: 12.07843
Float 64 bit: 2261634.50980392
Hexadecimal: 41 41 41 41
Decimal: 065 065 065 065
Octal: 101 101 101 101
Binary: 01000001 01000001 01000001
ASCII Text: AAAA

☐ Show little endian decoding ☐ Show unsigned as hexadecimal

Offset: 9021 / 13684 Selection: None INS

```
[01/02/22]seed@VM:~$ bless suffix
```



```
[01/02/22]seed@VM:~$ head -c 97 suffix > suffix_1
[01/02/22]seed@VM:~$ tail -c +225 suffix > suffix_2
[01/02/22]seed@VM:~$ ls
Desktop    file2      out2.bin  Public    suffix_2     task3.c    Videos
Documents  file2.txt  out.txt   q         task3_1      task3.o
Downloads  file3.txt  p         snap      task3_2      task4.c
file1      Music      Pictures  suffix    task3_a.bin  task4.o
file1.txt  out1.bin   prefix    suffix_1  task3_b.bin  Templates
[01/02/22]seed@VM:~$ cat prefix p suffix_1 p suffix_2 > task4_1
[01/02/22]seed@VM:~$ cat prefix q suffix_1 p suffix_2 > task4_2
[01/02/22]seed@VM:~$ 
```