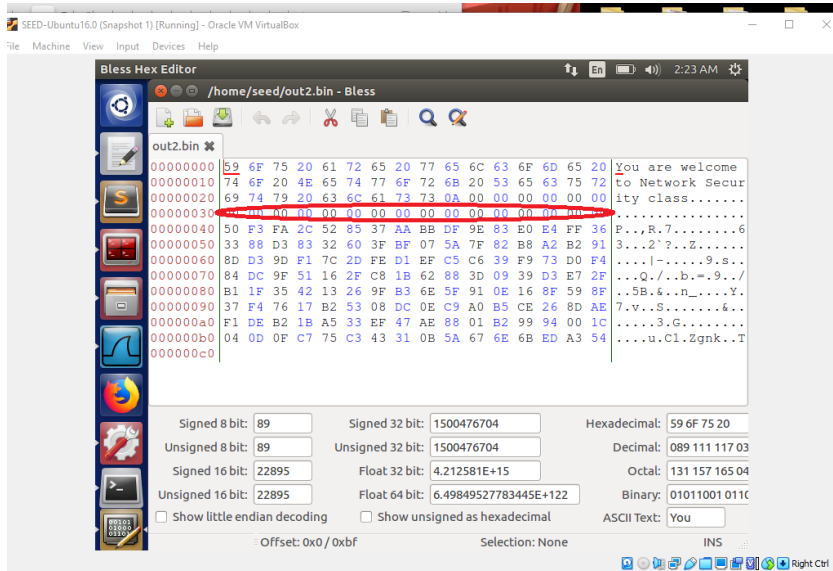


## Task 1.

### Generating Two Different Files with the same MD5 Hash

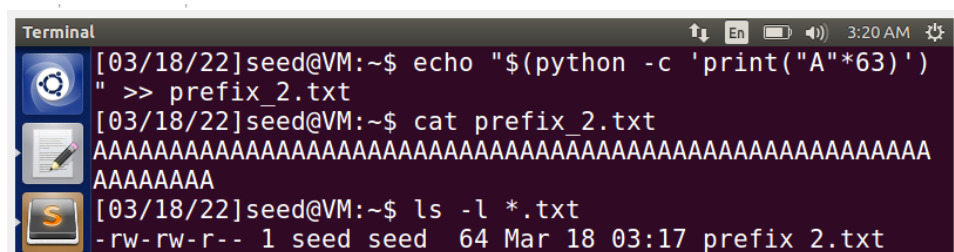
**Question 1.** If the length of your prefix file is not multiple of 64, what is going to happen?

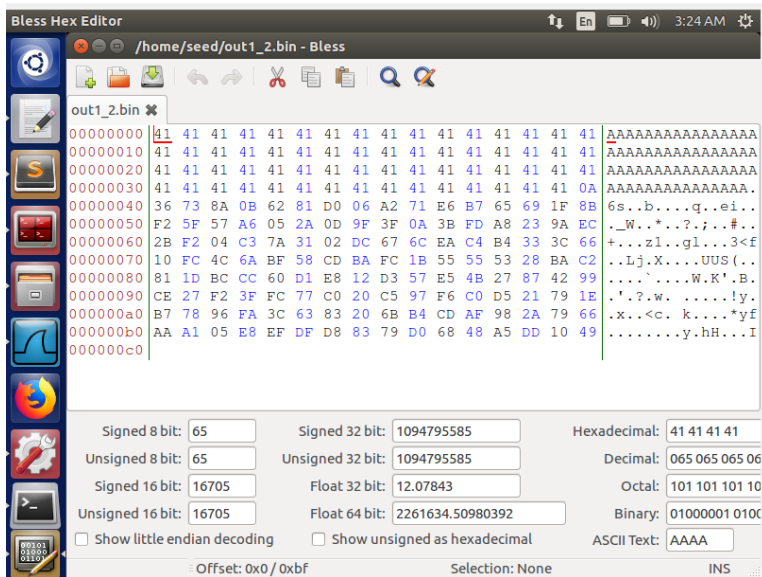
Nothing is going to happen as remnant bytes are always padded with zeros. See screen shot below.



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

No Padding at all.





**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

```

[03/18/22]seed@VM:~$ xxd out1_2.bin > p.txt
[03/18/22]seed@VM:~$ xxd out2_2.bin > q.txt
[03/18/22]seed@VM:~$ diff p.txt q.txt
6,8c6,8
< 00000050: f25f 57a6 052a 0d9f 3f0a 3bfd a823 9aec  .W...*...?.;...#..
< 00000060: 2bf2 04c3 7a31 02dc 676c eac4 b433 3c66  +...z1..gl...3<f
< 00000070: 10fc 4c6a bf58 cdba fc1b 5555 5328 bac2  ..Lj.X....UUS(..
---
> 00000050: f25f 5726 052a 0d9f 3f0a 3bfd a823 9aec  .W&.*...?.;...#..
> 00000060: 2bf2 04c3 7a31 02dc 676c eac4 b4b3 3c66  +...z1..gl....<f
> 00000070: 10fc 4c6a bf58 cdba fc1b 55d5 5328 bac2  ..Lj.X....U.S(..
10,12c10,12
< 00000090: ce27 f23f fc77 c020 c597 f6c0 d521 791e  .'?.w. ....!y.
< 000000a0: b778 96fa 3c63 8320 6bb4 cdaf 962a 7966  .x...<c. k....*yf
< 000000b0: aaa1 05e8 efd8 d883 79d0 6848 a5dd 1049  ....y.hH...I
---
> 00000090: ce27 f2bf fc77 c020 c597 f6c0 d521 791e  .'...w. ....!y.
> 000000a0: b778 96fa 3c63 8320 6bb4 cdaf 96aa 7866  .x...<c. k....xf
> 000000b0: aaa1 05e8 efd8 d883 79d0 68c8 a5dd 1049  ....y.h....I
[03/18/22]seed@VM:~$

```

## Task 2.

### Understanding MD5's Property

If  $MD5(M) = MD5(N)$ , Then  $MD5(M || T) = MD5(N || T)$ , where  $||$  represents concatenation for any input T

nal

```
[03/18/22]seed@VM:~$ echo "Network Security" >> file1.txt
[03/18/22]seed@VM:~$ echo "Network Security" >> file2.txt
[03/18/22]seed@VM:~$ md5sum file.txt
md5sum: file.txt: No such file or directory
[03/18/22]seed@VM:~$ md5sum file1.txt
8902e5d27f815a6c6c6cea67b2212bd1  file1.txt
[03/18/22]seed@VM:~$ md5sum file2.txt
8902e5d27f815a6c6c6cea67b2212bd1  file2.txt
[03/18/22]seed@VM:~$
```

```
Terminal File Edit View Search Terminal Help
[03/18/22]seed@VM:~$ echo "I love this course" >> file3.txt
[03/18/22]seed@VM:~$ md5sum file3.txt
51328aa92e3c5b662c8cb72c5f517148  file3.txt
[03/18/22]seed@VM:~$ cat file1.txt file3.txt > file_p
[03/18/22]seed@VM:~$ cat file2.txt file3.txt > file_q
[03/18/22]seed@VM:~$ md5sum file_p
3eebf290e8097cc9d2b4bfe99c1c1620  file_p
[03/18/22]seed@VM:~$ md5sum file_q
3eebf290e8097cc9d2b4bfe99c1c1620  file_q
[03/18/22]seed@VM:~$
```

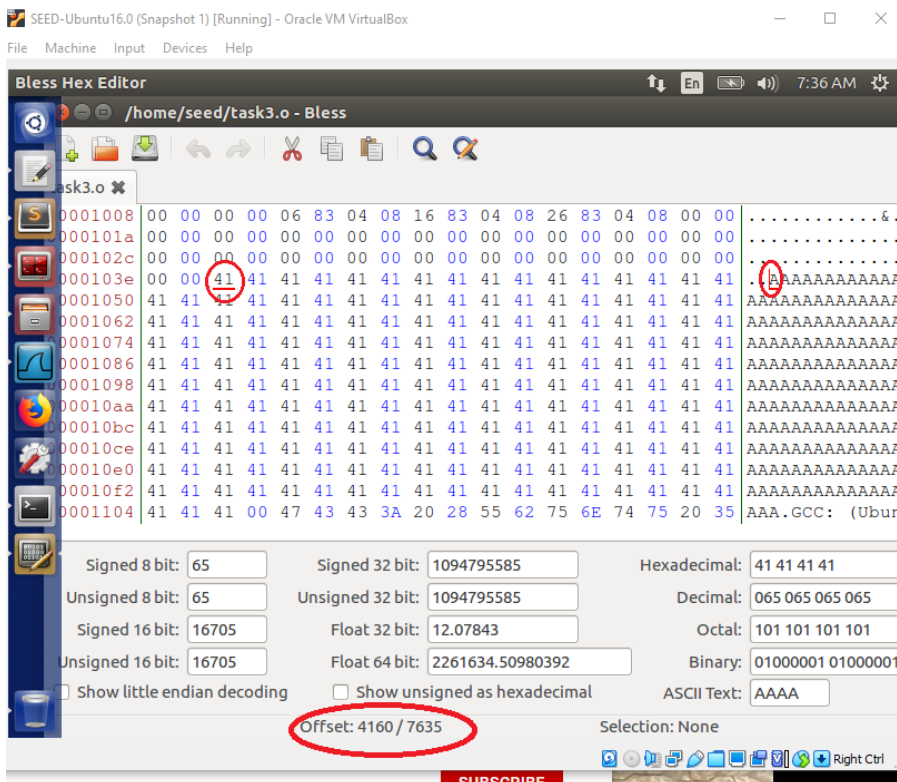
### Task 3.

Generating Two Executable Files with the Same MD5 Hash

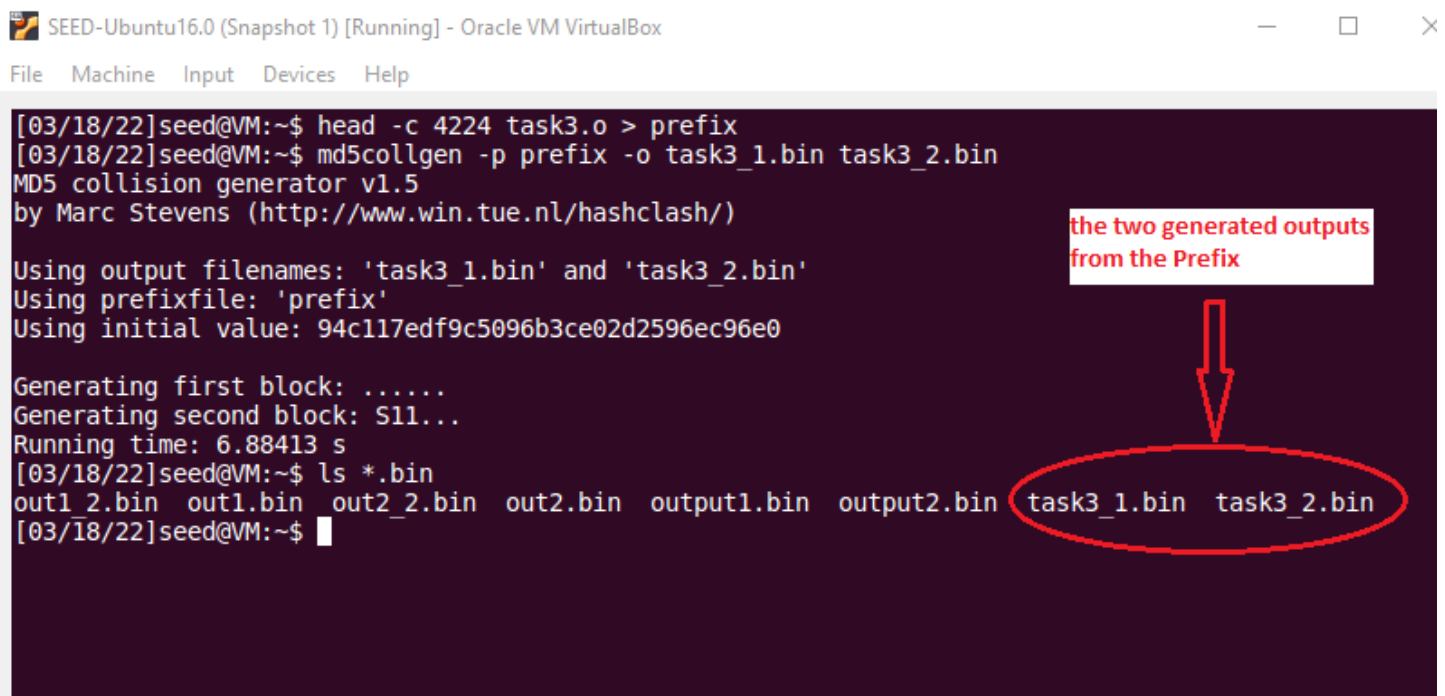
Creating fixed values – 0x40 into the file name outfile.txt and inserting them into the array of task3.c

[illegible][illegible]

From the start to the beginning of A is multiple of 64 - 4160



Since our prefix is from the start with some data from A,  
 We can say Prefix = 4160 + 64 (to maintain a multiple of 64).  
 Then we use the “head” command to get the first part of the file






Since we know that whole file contains the prefix + 128 bytes + suffix

We can use the tail command to get the last part – suffix

Then get p and q, and compare the two executable files (task3\_A & task3\_B) to be sure they both have the same MD5 hash.

**MD5 (prefix || P || suffix) = MD5 (prefix || Q || suffix)**

 SEED-Ubuntu16.0 (Snapshot 1) [Running] - Oracle VM VirtualBox

File Machine Input Devices Help

```
[03/18/22]seed@VM:~$ tail -c +4352 task3.o > suffix
[03/18/22]seed@VM:~$ tail -c 128 task3_1.bin > p
[03/18/22]seed@VM:~$ tail -c 128 task3_2.bin > q
[03/18/22]seed@VM:~$ cat prefix p suffix > task3_A
[03/18/22]seed@VM:~$ cat prefix q suffix > task3_B
[03/18/22]seed@VM:~$ md5sum task3_A
cc80434c55d2caf48b66e4785aff3a8a task3_A
[03/18/22]seed@VM:~$ md5sum task3_B
cc80434c55d2caf48b66e4785aff3a8a task3_B
[03/18/22]seed@VM:~$ █
```

**Task 4.**

## Making the Two Programs Behave Differently

### Creating the two program X[ ] and Y[ ]

[illegible]

```

root@seed:~# gcc main.c -o main
root@seed:~# ./main
x[0] = x[0] * x[0]
x[1] = x[1] * x[1]
x[2] = x[2] * x[2]
x[3] = x[3] * x[3]
x[4] = x[4] * x[4]
x[5] = x[5] * x[5]
x[6] = x[6] * x[6]
x[7] = x[7] * x[7]
x[8] = x[8] * x[8]
x[9] = x[9] * x[9]
x[10] = x[10] * x[10]
x[11] = x[11] * x[11]
x[12] = x[12] * x[12]
x[13] = x[13] * x[13]
x[14] = x[14] * x[14]
x[15] = x[15] * x[15]
x[16] = x[16] * x[16]
x[17] = x[17] * x[17]
x[18] = x[18] * x[18]
x[19] = x[19] * x[19]
x[20] = x[20] * x[20]
x[21] = x[21] * x[21]
x[22] = x[22] * x[22]
x[23] = x[23] * x[23]
x[24] = x[24] * x[24]
x[25] = x[25] * x[25]
x[26] = x[26] * x[26]
x[27] = x[27] * x[27]
x[28] = x[28] * x[28]
x[29] = x[29] * x[29]
x[30] = x[30] * x[30]
x[31] = x[31] * x[31]
x[32] = x[32] * x[32]
x[33] = x[33] * x[33]
x[34] = x[34] * x[34]
x[35] = x[35] * x[35]
x[36] = x[36] * x[36]
x[37] = x[37] * x[37]
x[38] = x[38] * x[38]
x[39] = x[39] * x[39]
x[40] = x[40] * x[40]
x[41] = x[41] * x[41]
x[42] = x[42] * x[42]
x[43] = x[43] * x[43]
x[44] = x[44] * x[44]
x[45] = x[45] * x[45]
x[46] = x[46] * x[46]
x[47] = x[47] * x[47]
x[48] = x[48] * x[48]
x[49] = x[49] * x[49]
x[50] = x[50] * x[50]
x[51] = x[51] * x[51]
x[52] = x[52] * x[52]
x[53] = x[53] * x[53]
x[54] = x[54] * x[54]
x[55] = x[55] * x[55]
x[56] = x[56] * x[56]
x[57] = x[57] * x[57]
x[58] = x[58] * x[58]
x[59] = x[59] * x[59]
x[60] = x[60] * x[60]
x[61] = x[61] * x[61]
x[62] = x[62] * x[62]
x[63] = x[63] * x[63]
x[64] = x[64] * x[64]
x[65] = x[65] * x[65]
x[66] = x[66] * x[66]
x[67] = x[67] * x[67]
x[68] = x[68] * x[68]
x[69] = x[69] * x[69]
x[70] = x[70] * x[70]
x[71] = x[71] * x[71]
x[72] = x[72] * x[72]
x[73] = x[73] * x[73]
x[74] = x[74] * x[74]
x[75] = x[75] * x[75]
x[76] = x[76] * x[76]
x[77] = x[77] * x[77]
x[78] = x[78] * x[78]
x[79] = x[79] * x[79]
x[80] = x[80] * x[80]
x[81] = x[81] * x[81]
x[82] = x[82] * x[82]
x[83] = x[83] * x[83]
x[84] = x[84] * x[84]
x[85] = x[85] * x[85]
x[86] = x[86] * x[86]
x[87] = x[87] * x[87]
x[88] = x[88] * x[88]
x[89] = x[89] * x[89]
x[90] = x[90] * x[90]
x[91] = x[91] * x[91]
x[92] = x[92] * x[92]
x[93] = x[93] * x[93]
x[94] = x[94] * x[94]
x[95] = x[95] * x[95]
x[96] = x[96] * x[96]
x[97] = x[97] * x[97]
x[98] = x[98] * x[98]
x[99] = x[99] * x[99]

```

```
};

int main()
{
    int i=0;
    for (i=0; i<400; i++){

        if(x[i]!=y[i]) break;
    }

    if(i==400){
        printf("GOOD\n");
    }
    else{
        printf("BAD\n");
    }
    return 0;
}
```

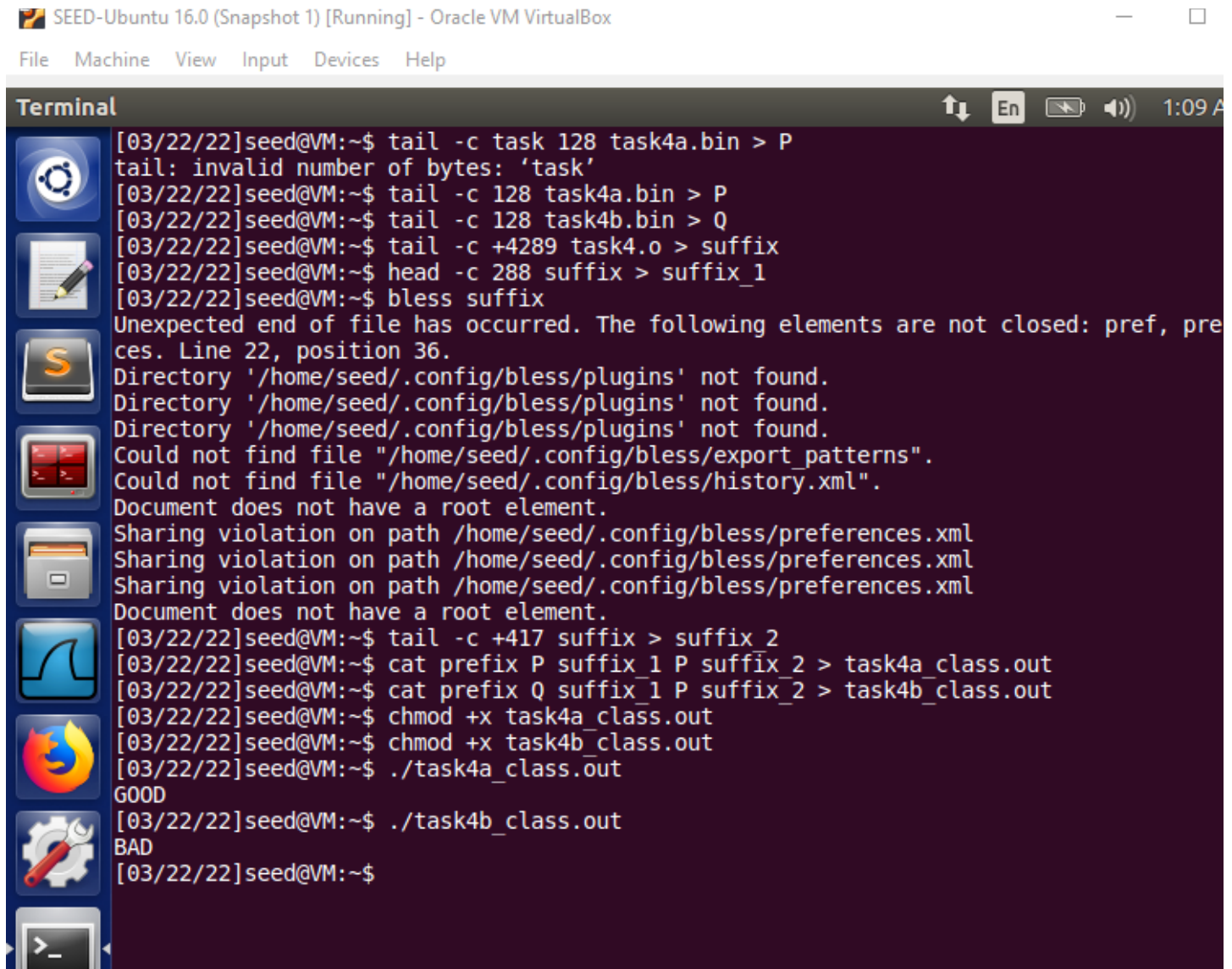
Using the previous knowledge of task 3, the prefix was gotten.







4289= (length of prefix) 4160+ (length of P) 128 + 1



```
SEED-Ubuntu 16.0 (Snapshot 1) [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help

Terminal
[03/22/22]seed@VM:~$ tail -c task 128 task4a.bin > P
tail: invalid number of bytes: 'task'
[03/22/22]seed@VM:~$ tail -c 128 task4a.bin > P
[03/22/22]seed@VM:~$ tail -c 128 task4b.bin > Q
[03/22/22]seed@VM:~$ tail -c +4289 task4.o > suffix
[03/22/22]seed@VM:~$ head -c 288 suffix > suffix_1
[03/22/22]seed@VM:~$ bless suffix
Unexpected end of file has occurred. The following elements are not closed: pref, pre
ces. Line 22, position 36.
Directory '/home/seed/.config/bless/plugins' not found.
Directory '/home/seed/.config/bless/plugins' not found.
Directory '/home/seed/.config/bless/plugins' not found.
Could not find file "/home/seed/.config/bless/export_patterns".
Could not find file "/home/seed/.config/bless/history.xml".
Document does not have a root element.
Sharing violation on path /home/seed/.config/bless/preferences.xml
Sharing violation on path /home/seed/.config/bless/preferences.xml
Sharing violation on path /home/seed/.config/bless/preferences.xml
Document does not have a root element.
[03/22/22]seed@VM:~$ tail -c +417 suffix > suffix_2
[03/22/22]seed@VM:~$ cat prefix P suffix_1 P suffix_2 > task4a_class.out
[03/22/22]seed@VM:~$ cat prefix Q suffix_1 P suffix_2 > task4b_class.out
[03/22/22]seed@VM:~$ chmod +x task4a_class.out
[03/22/22]seed@VM:~$ chmod +x task4b_class.out
[03/22/22]seed@VM:~$ ./task4a_class.out
GOOD
[03/22/22]seed@VM:~$ ./task4b_class.out
BAD
[03/22/22]seed@VM:~$
```

The results were confirmed by running the md5sum and diff command on the 2 resultant outputs.

## OBSERVATIONS

After carrying out this project, it was noticed that two executable files can be generated with the same MD5 hash. And these two files or programs can be manipulated to behave in different ways.

It was also observed that the concatenation property of the MD5, for any input T,  $MD5(M \parallel T) = MD5(N \parallel T)$ .