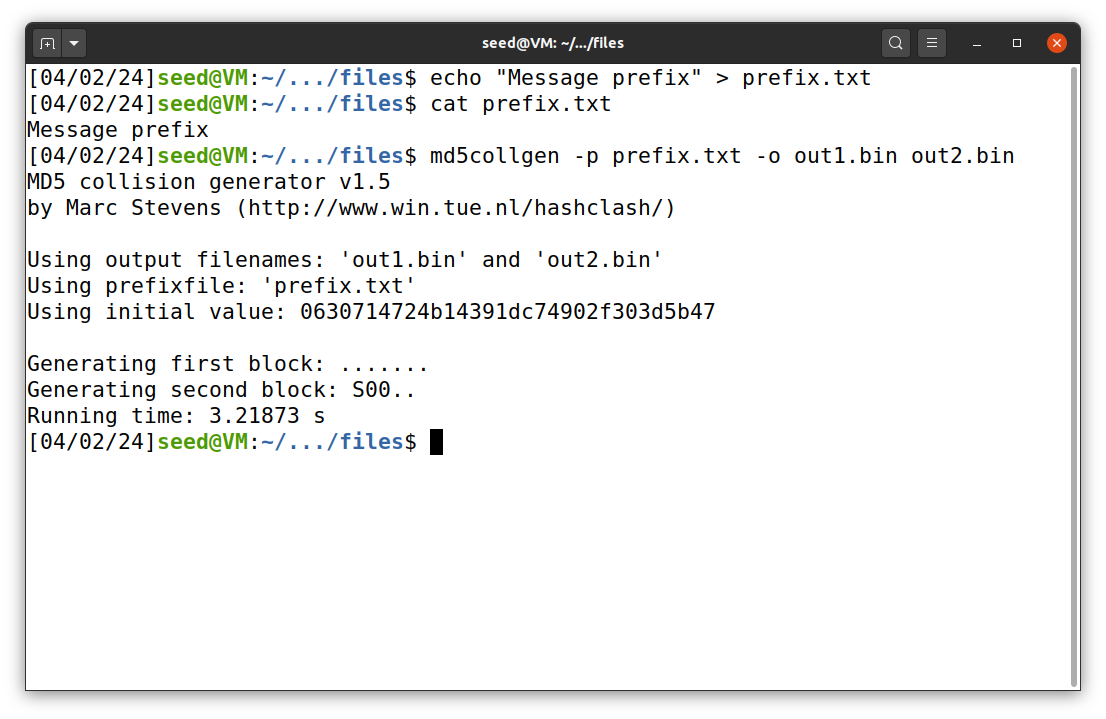
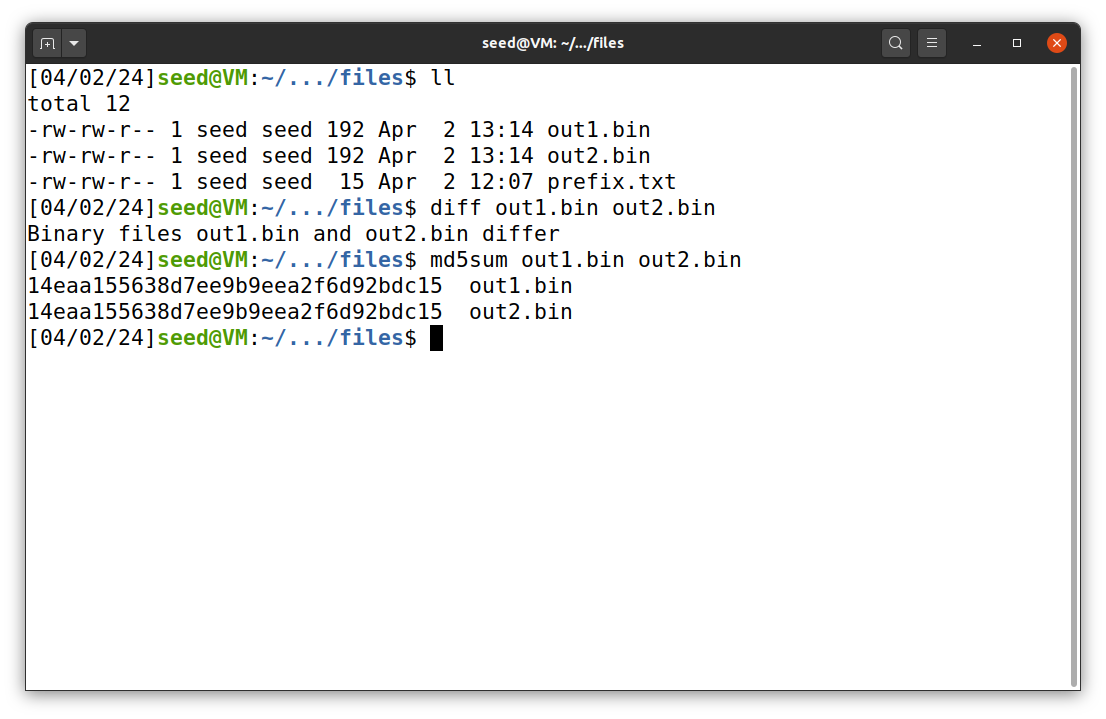
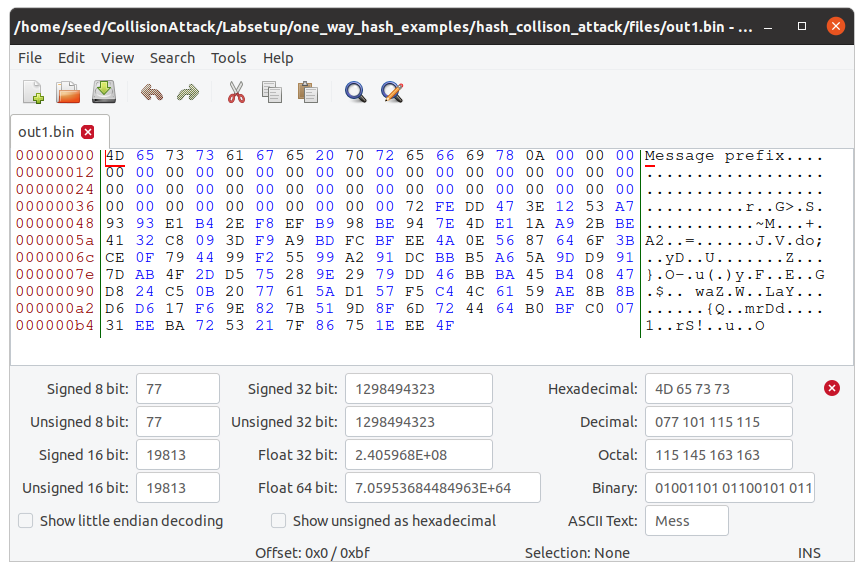
**Lab 9 MD 5 Collision**

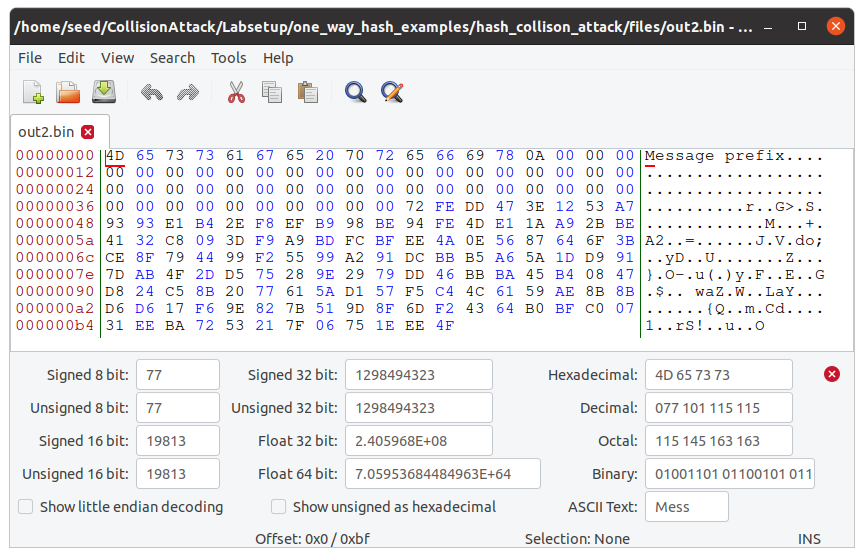
**Task 1: Generating Two Different Files with Same MD5 Hash**

In this task we need to generate two different files with same MD5 hash values. First we create a prefix text file.

Then we run above command to generate two output files – out1.bin & out2.bin from the prefix file.

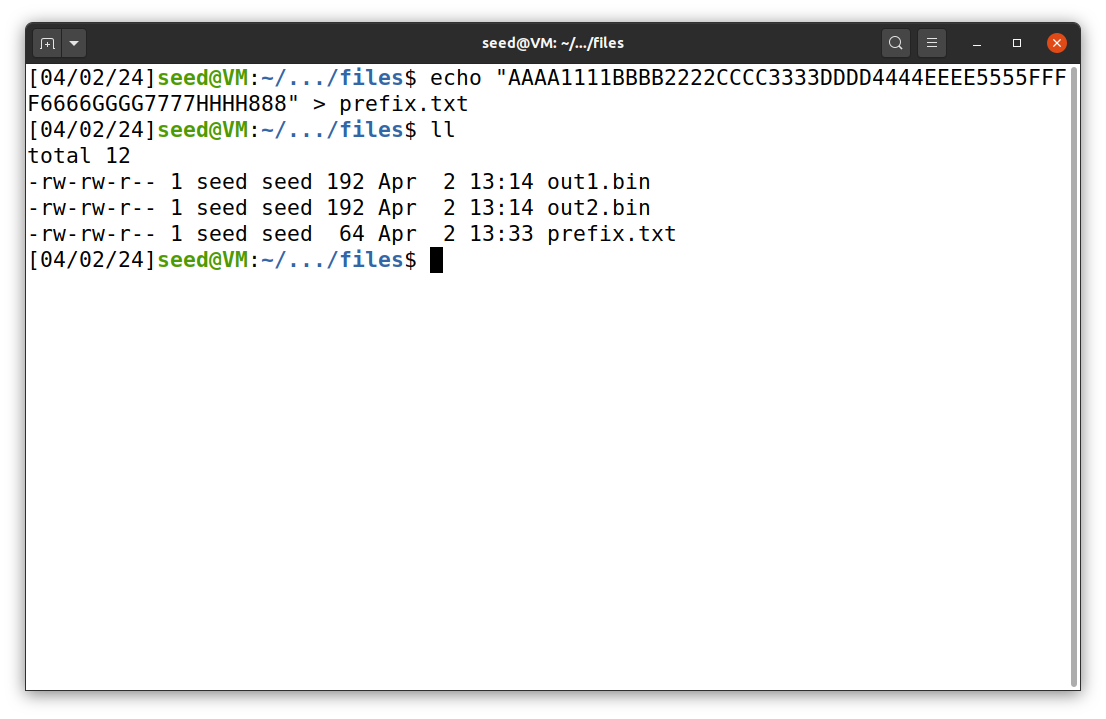
Using the diff command we check whether the output file are different or not. So as you can see they are different. Then me use the md5sum to check the MD5 hash of both files.

Using the bless command we can see the output file, this is out1.bin

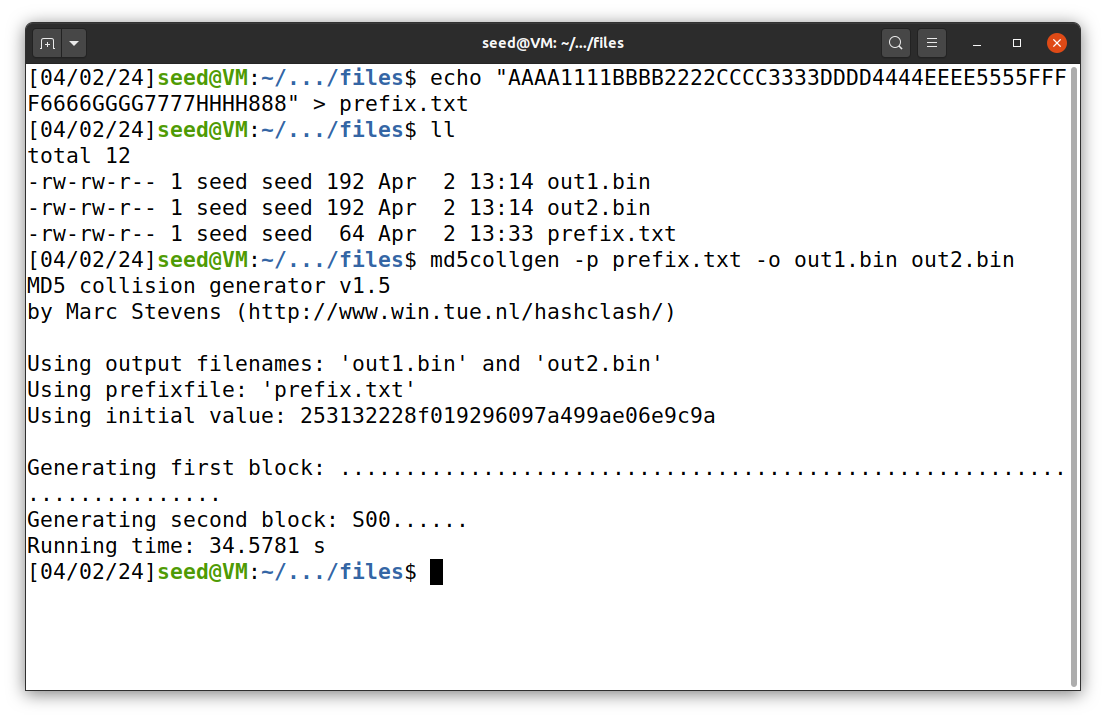
This is the out2.bin using the bless command.  
  
**Question 1- If the length of your prefix file is not multiple of 64, what is going to happen?**

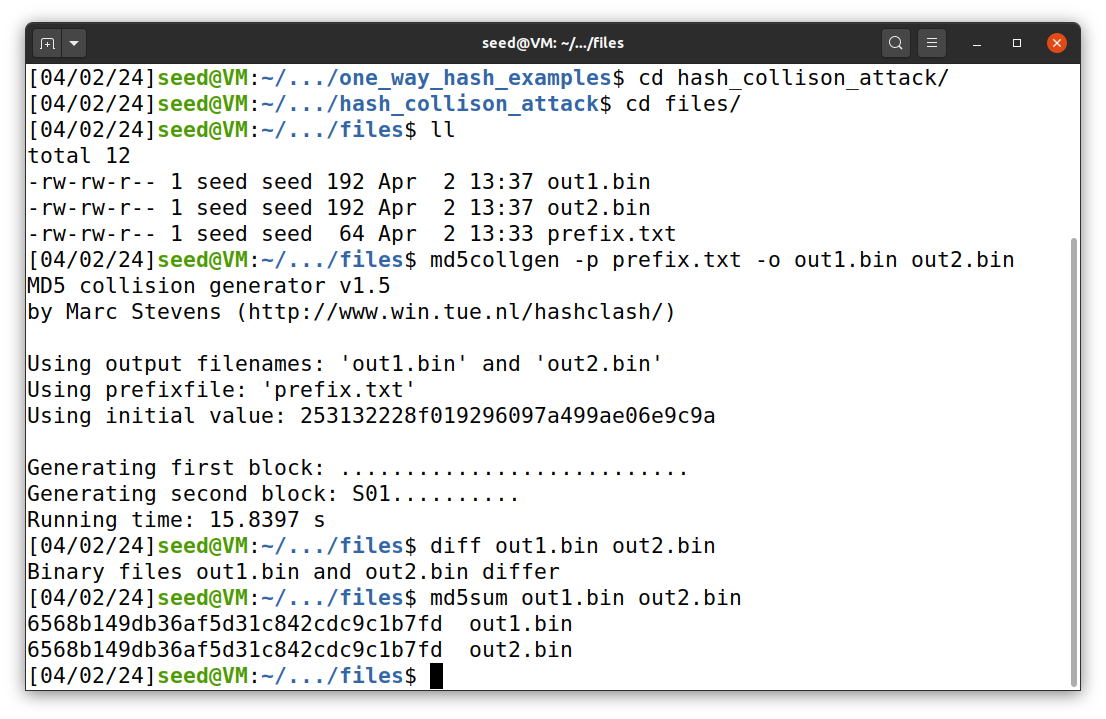
→ If the length of the prefix file is not multiple of 64 then the md5collgen will add in padding to make it to length of multiple of 64.

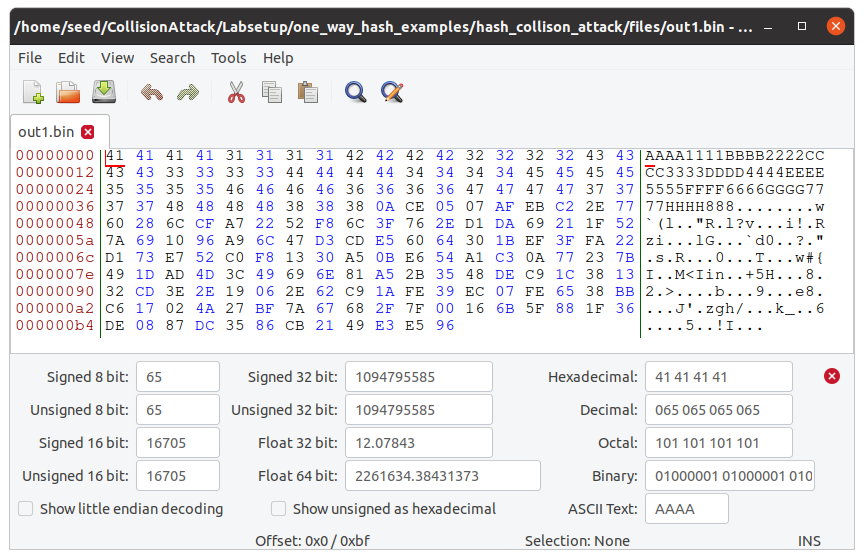
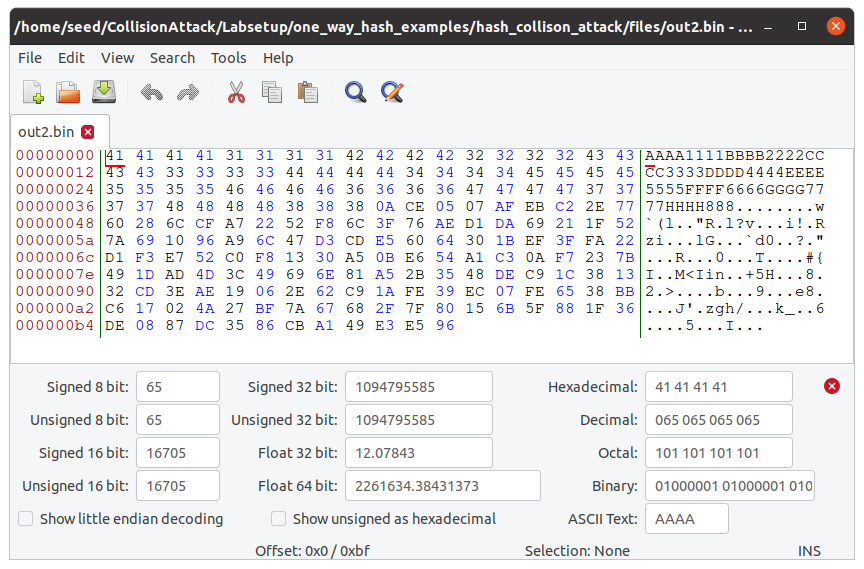
**Question 2 - Create a prefix file with exactly 64 bytes, and run the collision tool again, and see what happens**

**→**

So first we create a prefix text file of 64 bytes exactly, as you can see from the screenshot we created that.

Now we run the same commands that we run before to generate the the two output files – out1.bin & out2.bin

As you can see after using the diff command we get that the two output files differ and we also check the MD% hash which is similar.

This is out1.bin   
  
This is out2.bin

**Question 3 - Are the data (128 bytes) generated by md5collgen completely different for the two**

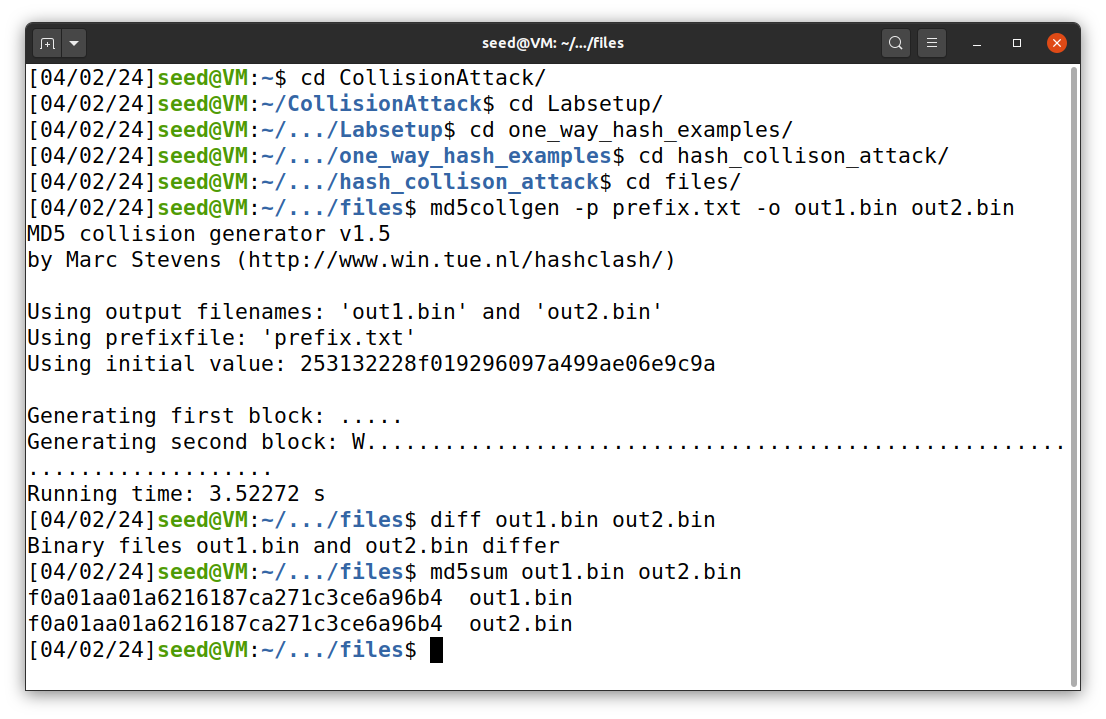
**output files? Please identify all the bytes that are different.**

**→** They are somewhat similar but not completely same. Some of the bytes are different.

Like the out1.bin has 4E, 37, 63, 95, 02, 40, F1 these bytes different from out2.bin

Out2.bin has these bytes at that position CE, B7, E3, 15, 82, C0, F0

**Task 2 : Understanding MD5’s Property**

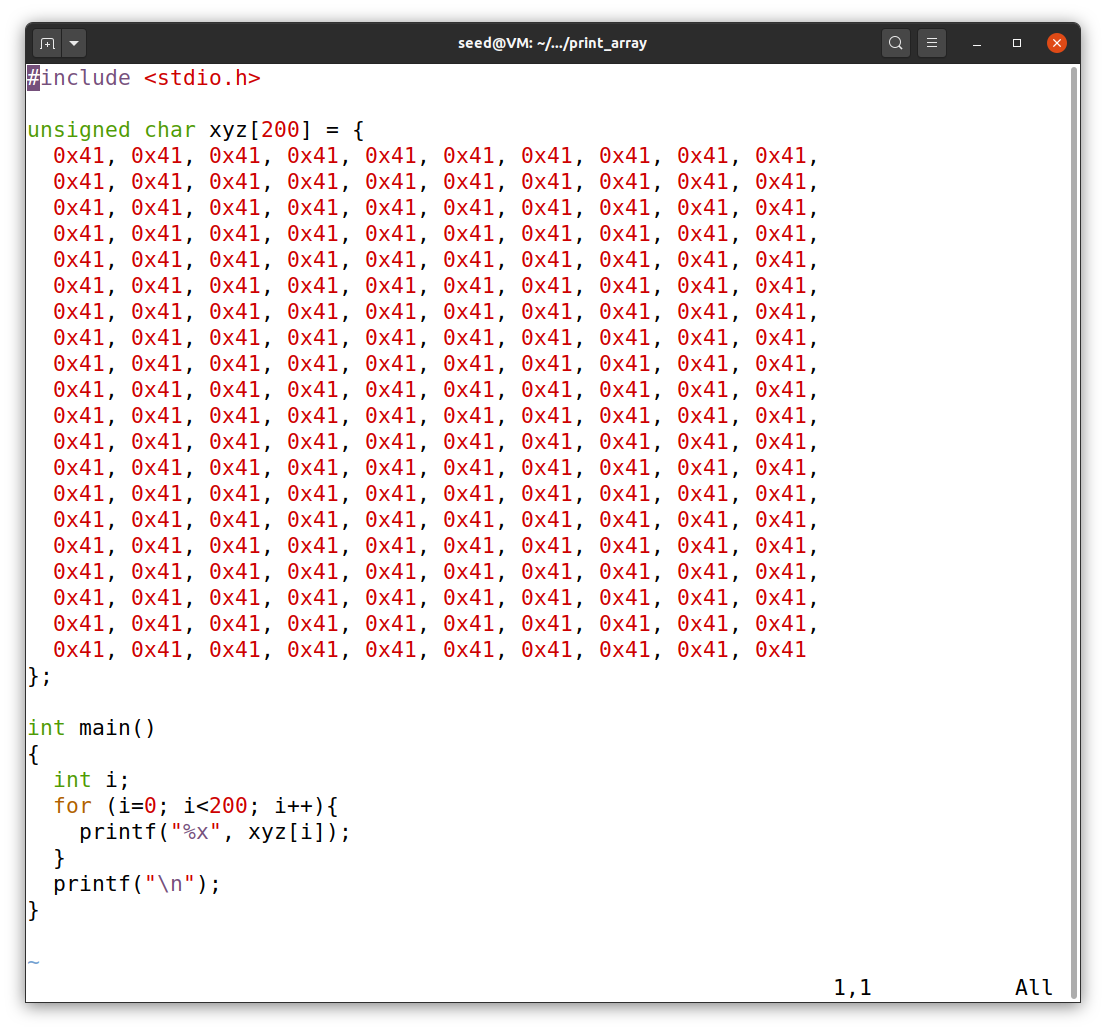
****So in this task we once again generate the two output files similarly how we did in task 1. And do the same operations we did in task 1.

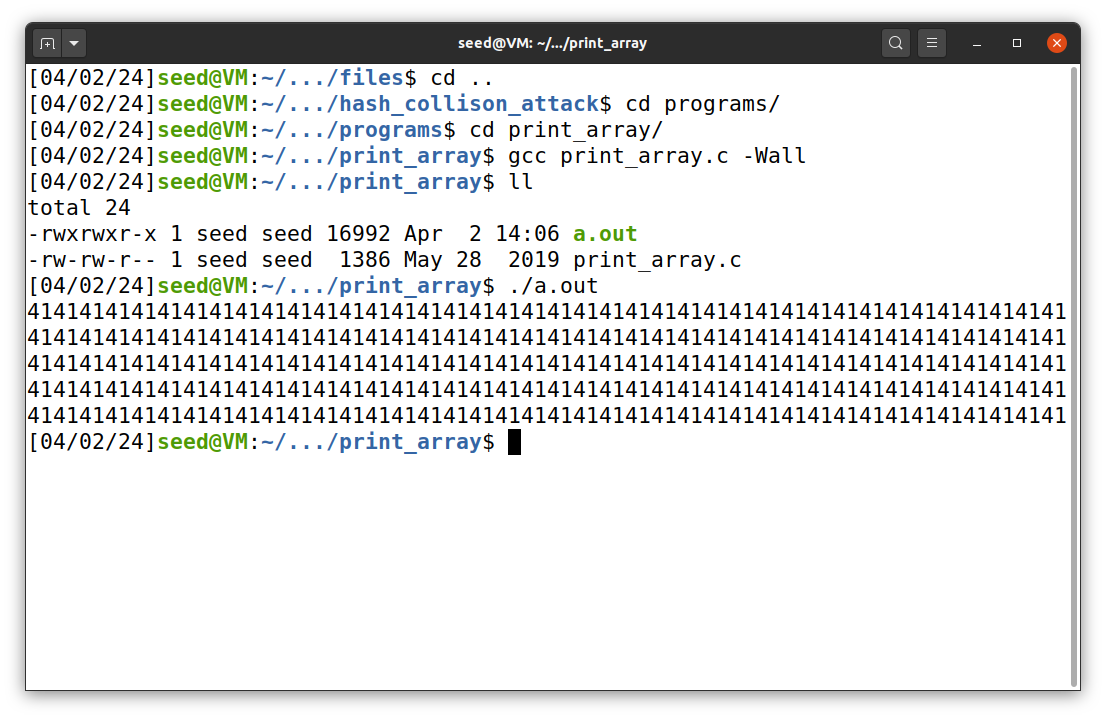
****After that using the echo command we generated new suffix text file and then using the cat command we concatenate the suffix text file to the end of both out1.bin and out2.bin to get new output files.

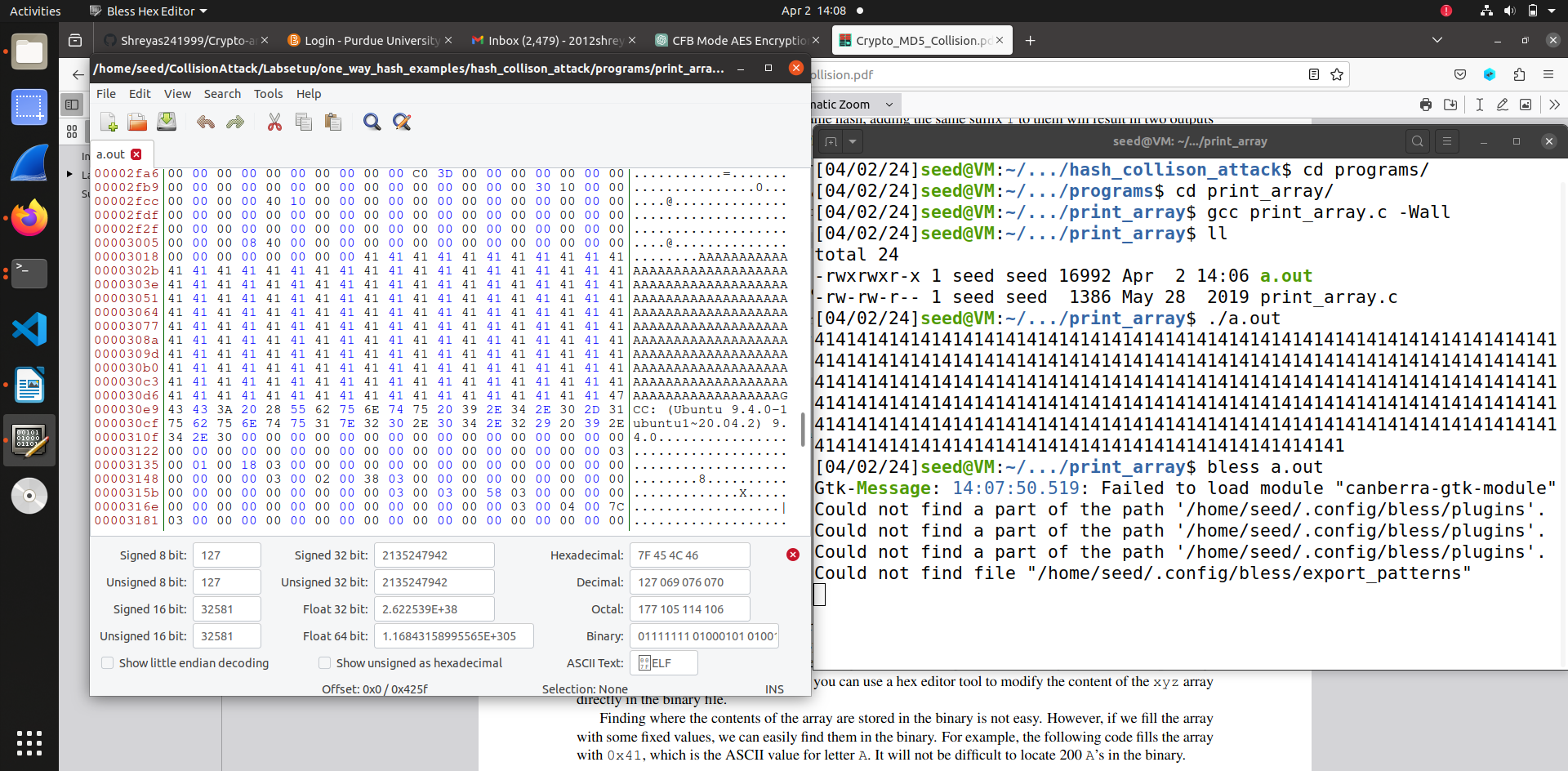
After that we use the same MD5 sum hash for both files to get same similar hash value.

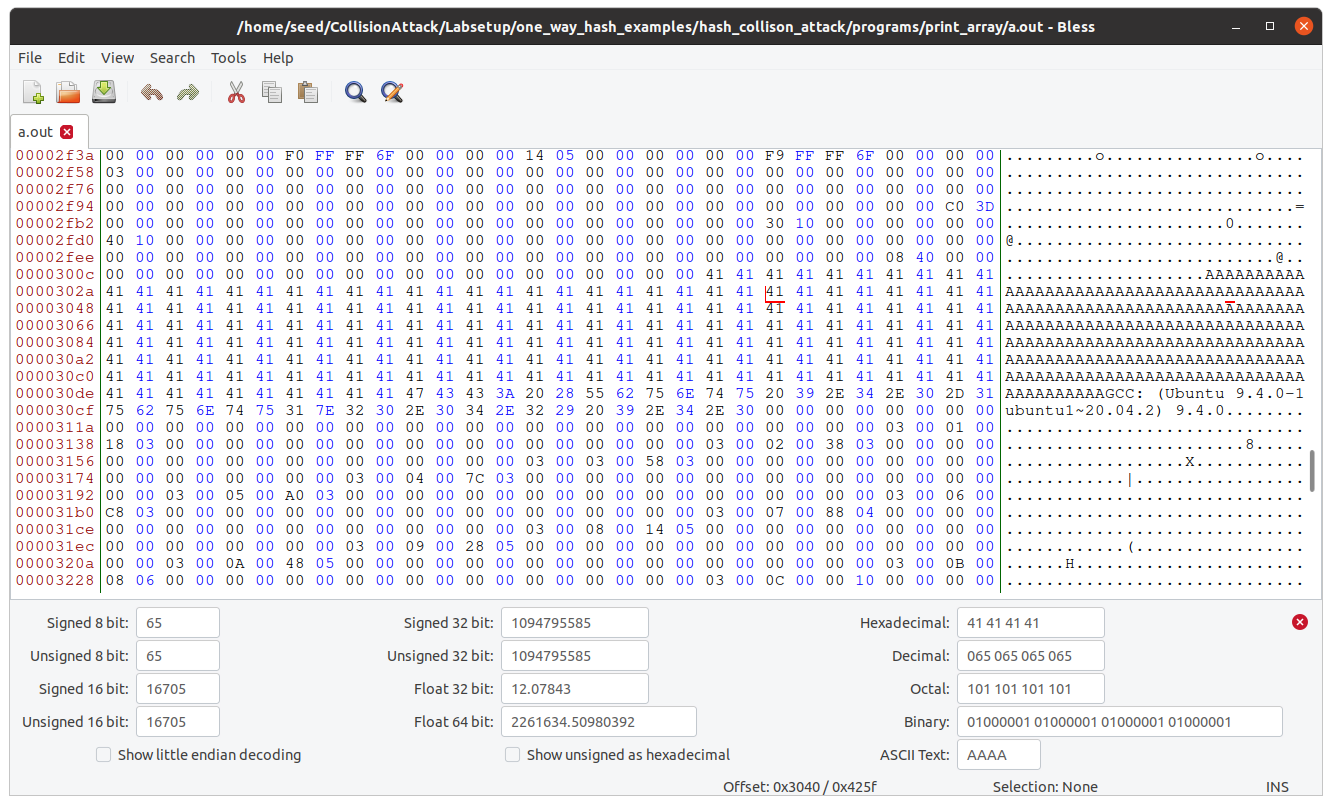
This proves that they are indeed same.

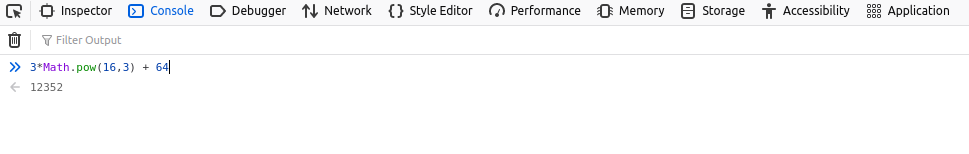
**Task 3: Generating Two Executable Files with Same MD5 Hash**

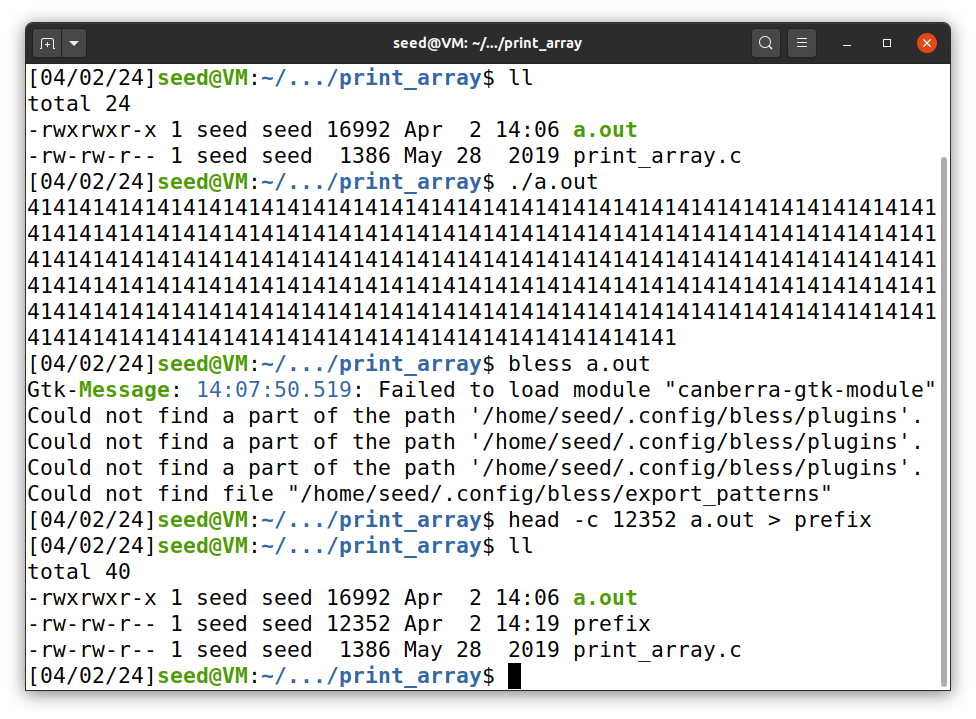
In this task using the above program we creare two different version of it, in which their content is different but the hash value is same.

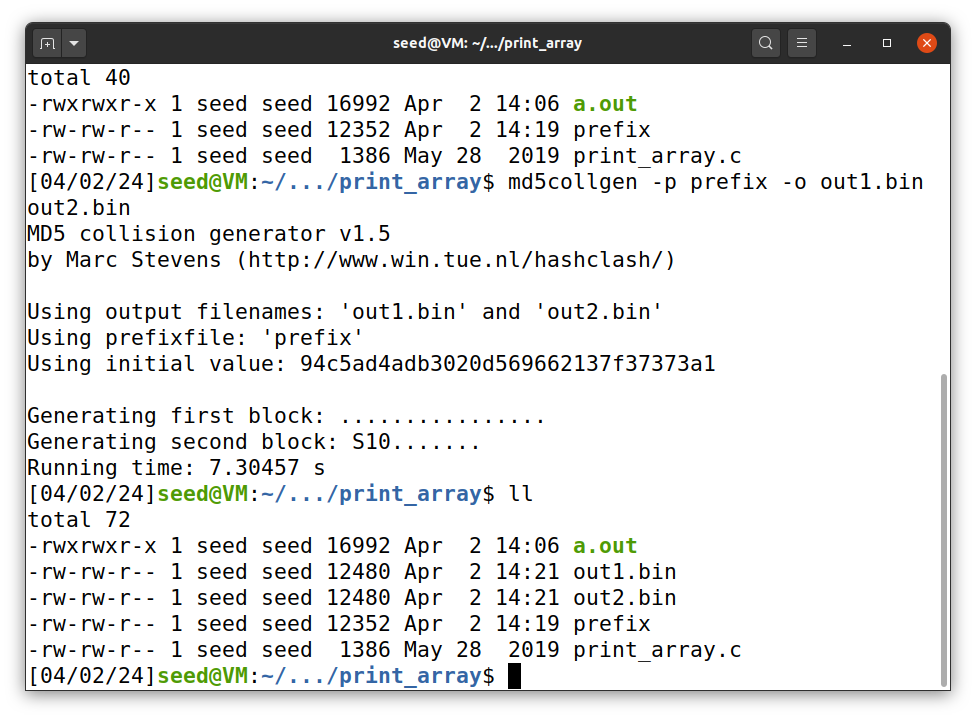
First we compile the c code using gcc and get the a.out file. When we run the a.out we get this output.

Then we use the bless command to get the content of the a.out file.

From the above bless file we get the necessary offest values that is 0x3040.

This is how we get the length of prefix.

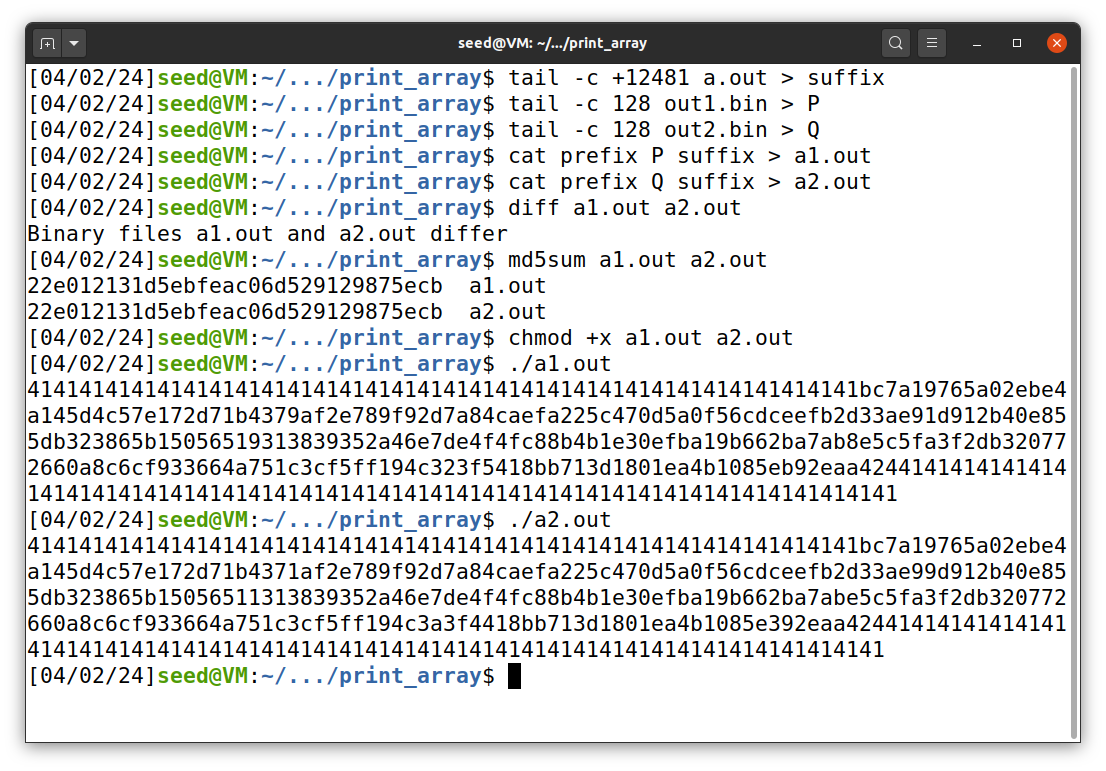
After that using head command we generate the prefix file.

Then after that using md5collgen to get the two output files that is out1.bin and out2.bin. When we run the ll command we can see that binary files are of size 12480, the prefix of size 12352, so from this we can start the suffix from 12481.

After that we generate the suffix file and generate P & Q .

Now we concatenate the prefix and suffix with P and Q .

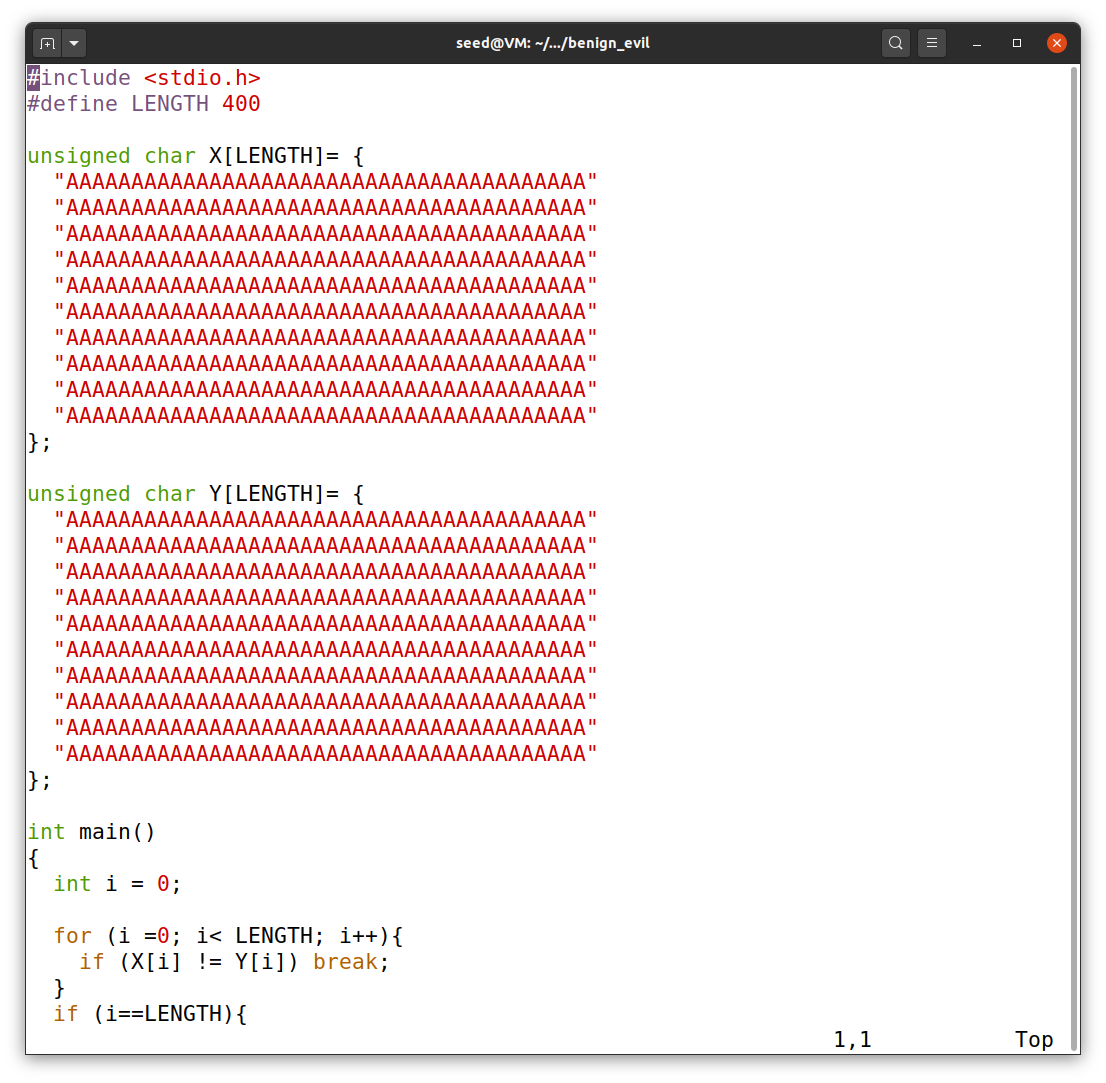
After that we give permissions to the both files – a1.out and a2.out and then check whether that are same or not , as you can see from the screenshot they differ and when we run the MD5sum both hash values are similar.

When we display both files this is the output we get.

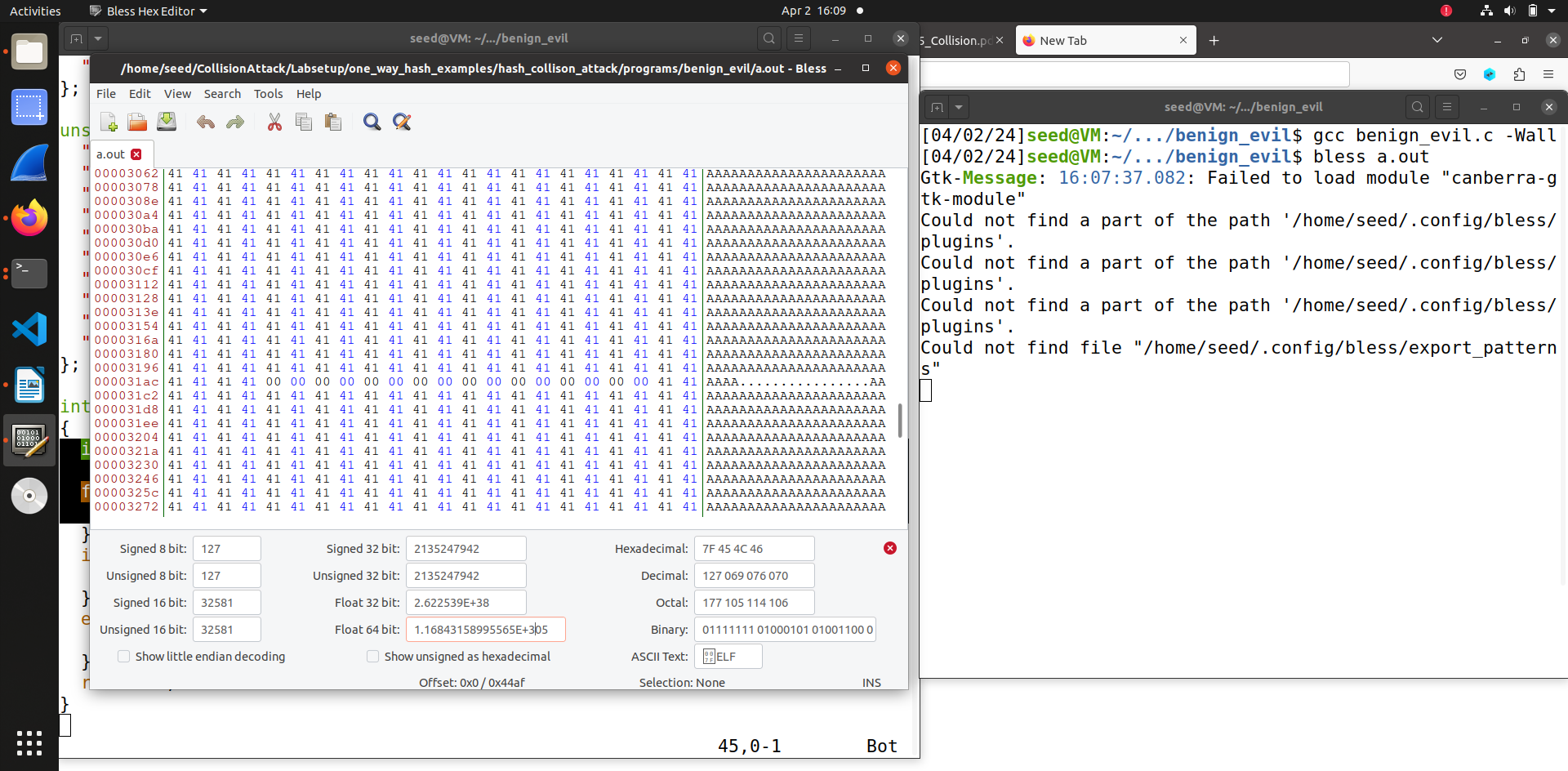
Thus we have successfully created two different versions from the C program with different content but same hash value.

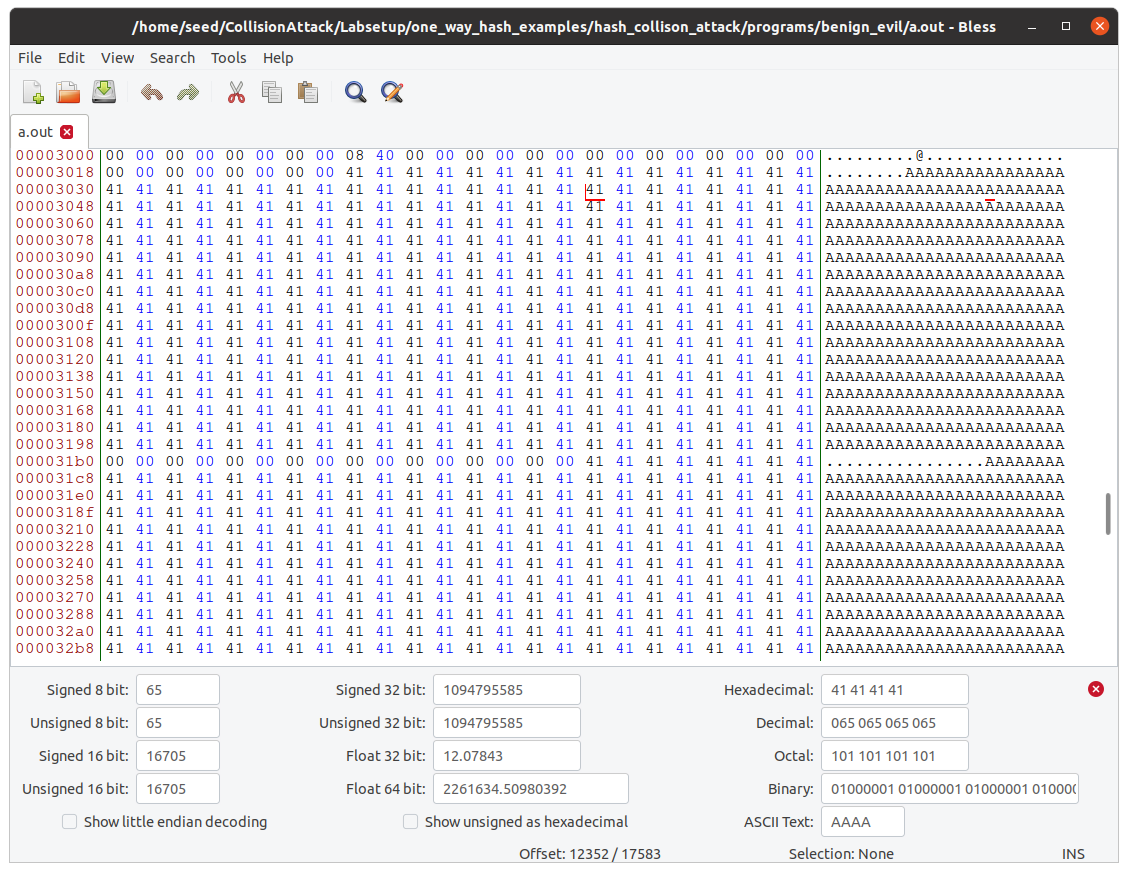
**Task 4 : Making the Two Programs Behave Differently**

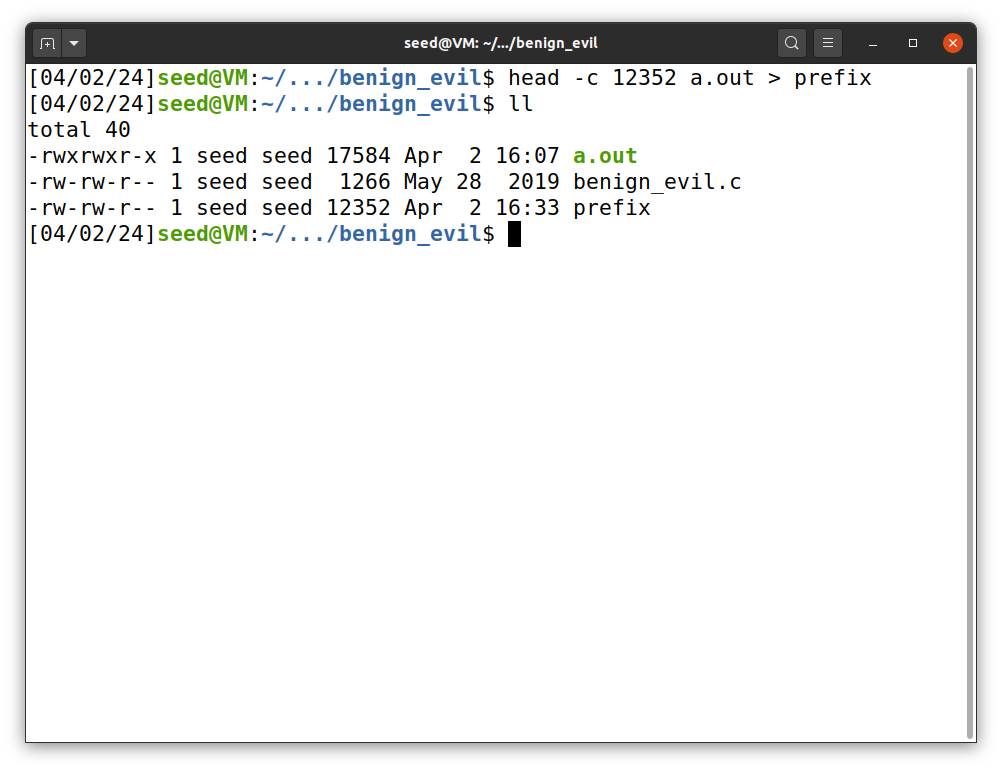
In this task using the below program we generate two versions of same program array with same hash values. We create two programs that share the same MD5 hash, but one program will execute benign instructions and the other will execute malicious instructions.

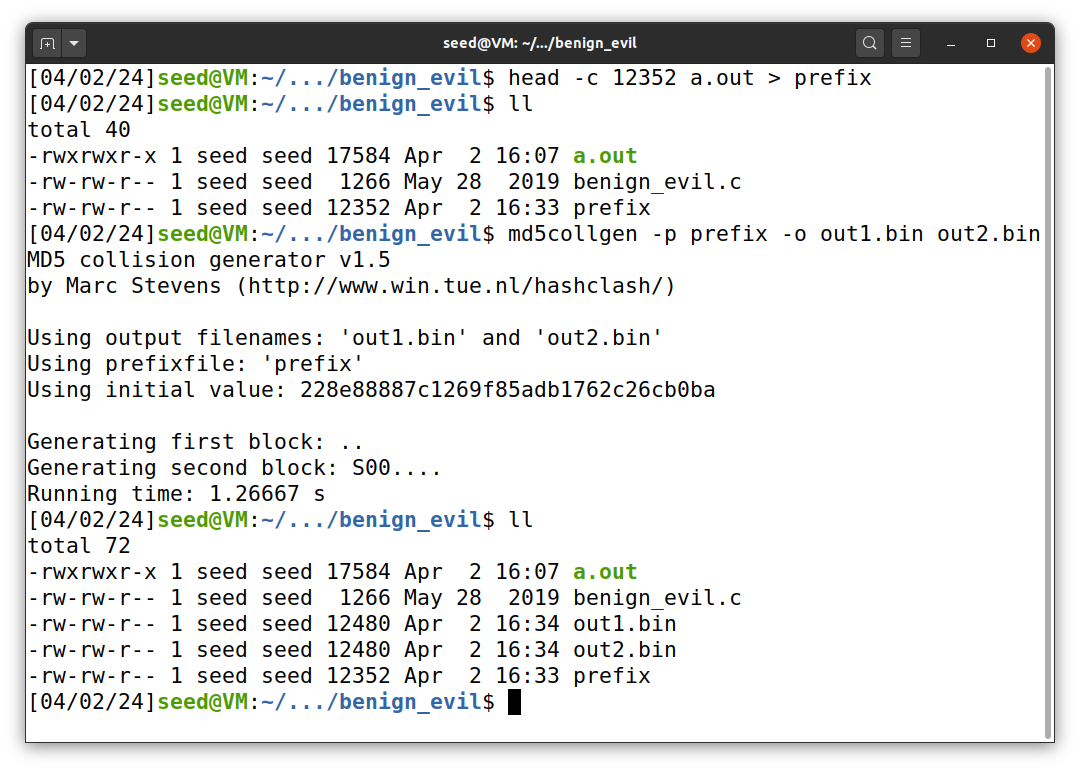


This is the code we will use for this task.

The first we do in this task is compile the program and then run the bless command on a.out.

Then we choose the position that is multiple of 64, so as you can see from the screenshot I have choosen 12352. We will use these first 12352 bytes of a.out to copy into prefix file.

So we copied the 12352 bytes into prefix file and run the ll command to check it.

Then we use MD5collgen to generate two output files out1.bin and out2.bin, which would be the prefix + P and prefix + Q . And when ll when we get the binary files have a size of 12480 bytes and the prefix file has size of 12352

After that we use the tail command to do 12352 + 128 to get 12480

The suffix array has suffux\_1, P/Q and suffix\_2. So we overwrite the first 288 bytes with suffix\_1 and everything and the first of 417 bytes with suffix\_2  
Then we concatenated the files together into 2 output files. The first file has prefix + P + suffix\_1 + P + suffix\_2 and the second has prefix + Q + suffix\_1 + P + suffix\_2

After that we give permission to execute both files. As you can see from the screenshot one executes benign code and other executes malicious code. We also compare both files using diff command and also check the MD5sum hash value which is similar.   
  
Thus we have successfully executed the attack.