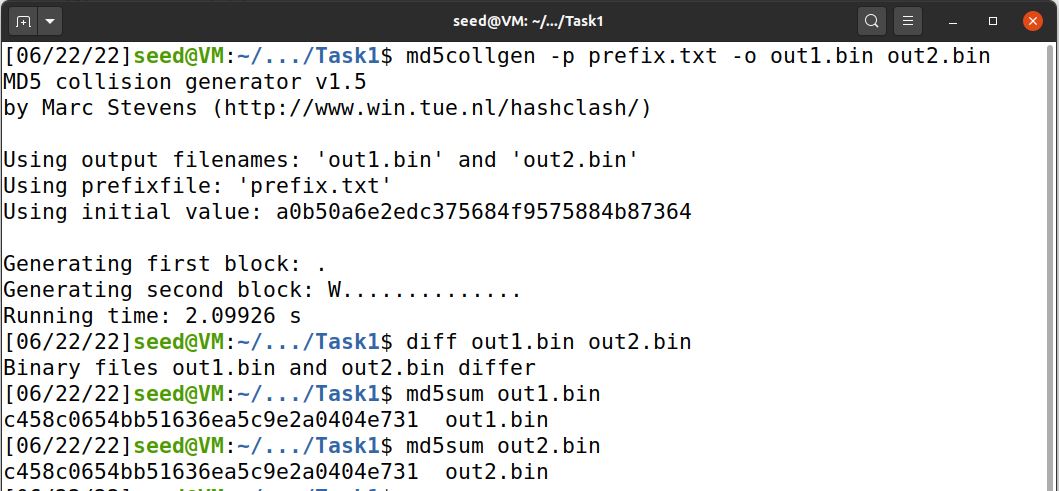
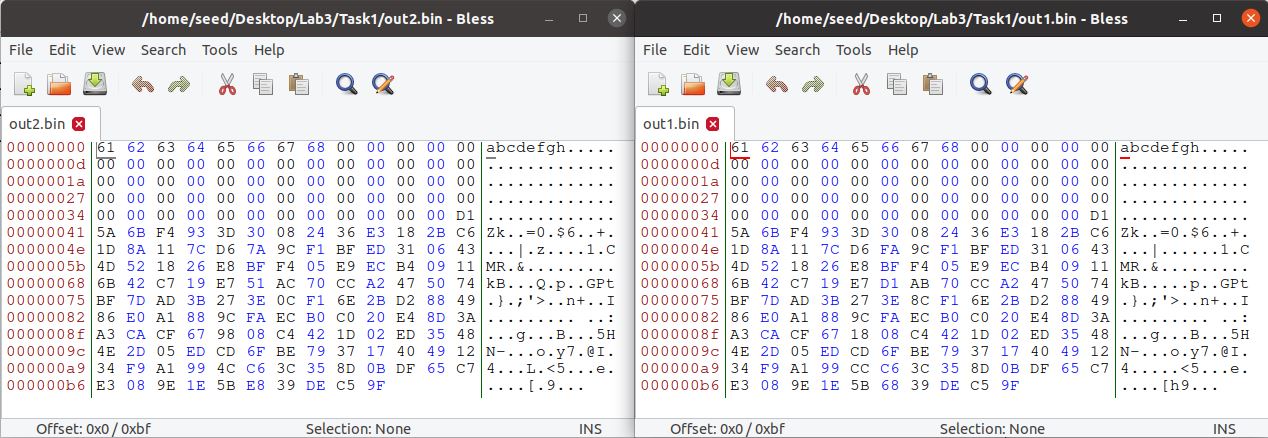
**Task 1: Generating Two Different Files with the Same MD5 Hash**

Initial few steps:

1. Creating files using prefix.txt whose value is “abcdefgh”.
2. Checking difference between them



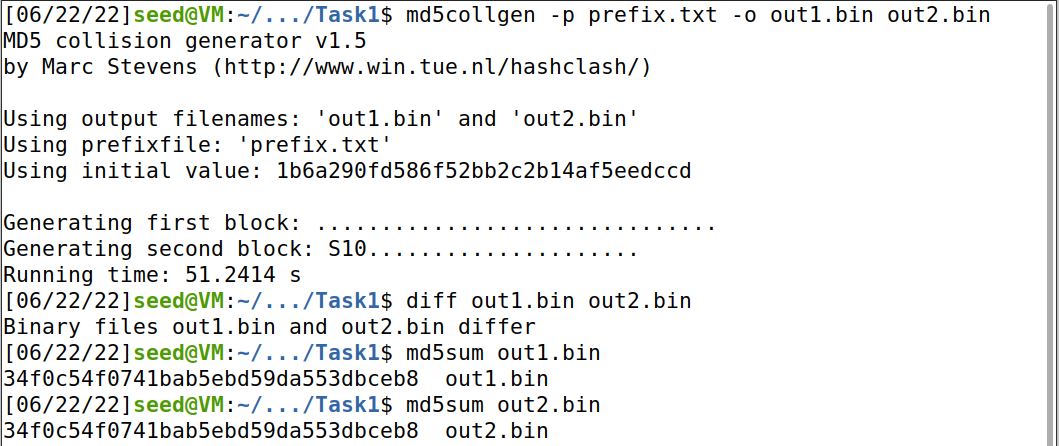
1. Showing each of the files using ‘bless’ editor



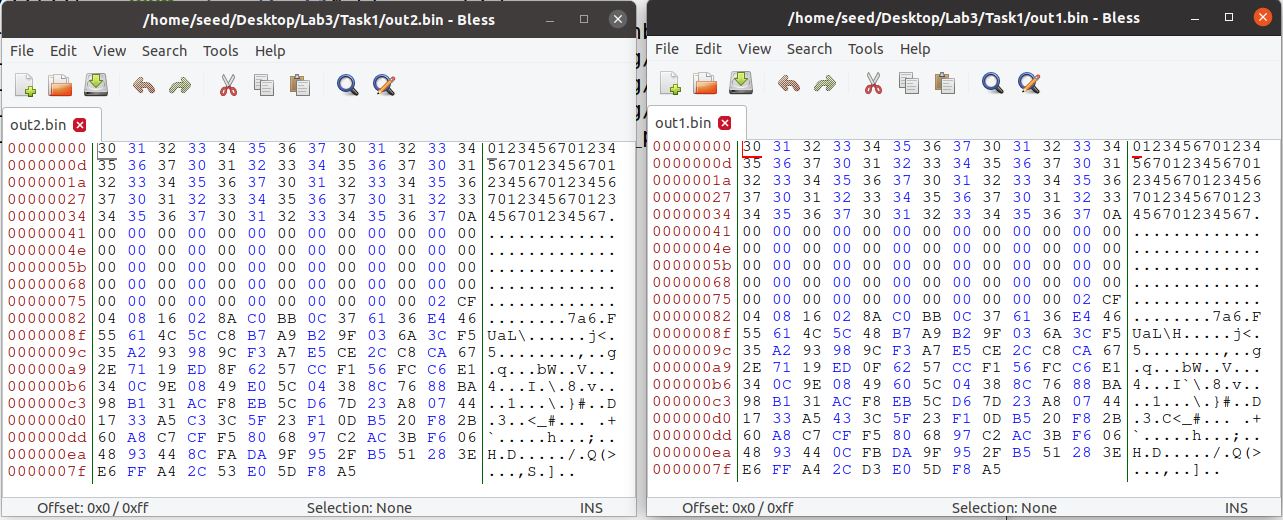
**Ques1.** When the prefix file is not in the multiple of 64, it will be padded with values such as '1' or '0' like the one above with only 5Bytes value.

**Ques2.**

Initial code :

****

Bless editor :



No padding is observed amongst the two files.

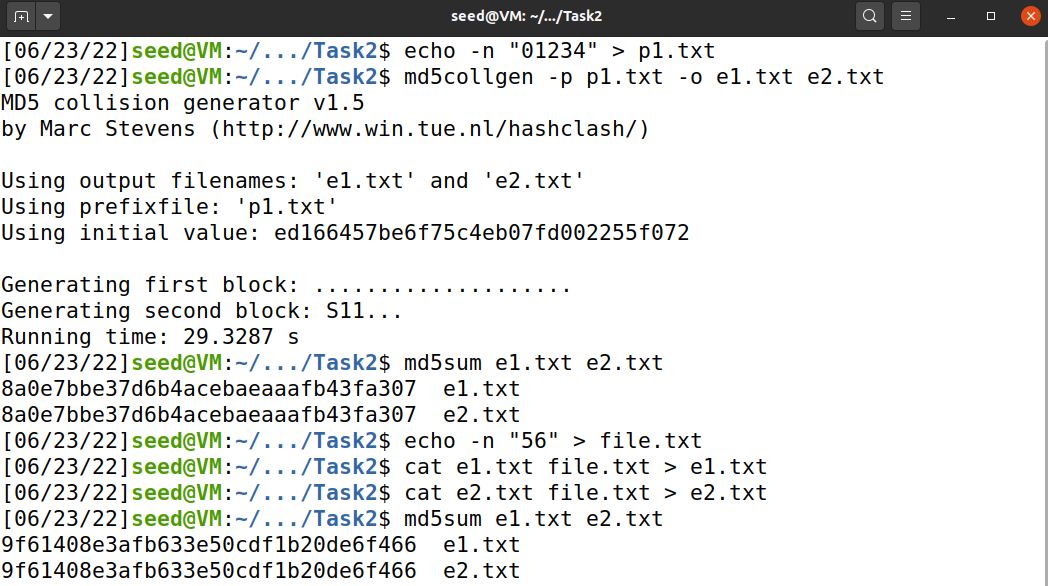
**Ques3.**

|  | What we observe here, that prefix file and padding consist of same particular data. The only difference is of P and Q which are differences like ‘Ws’ and ‘r’ |
| --- | --- |

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

After applying the following steps:

1. Creating two files with MD5 hash
2. Checking whether they have the same Hash values.
3. Creating a new file
4. Concatenating the new file as a suffix
5. Checking whether the hash values changed

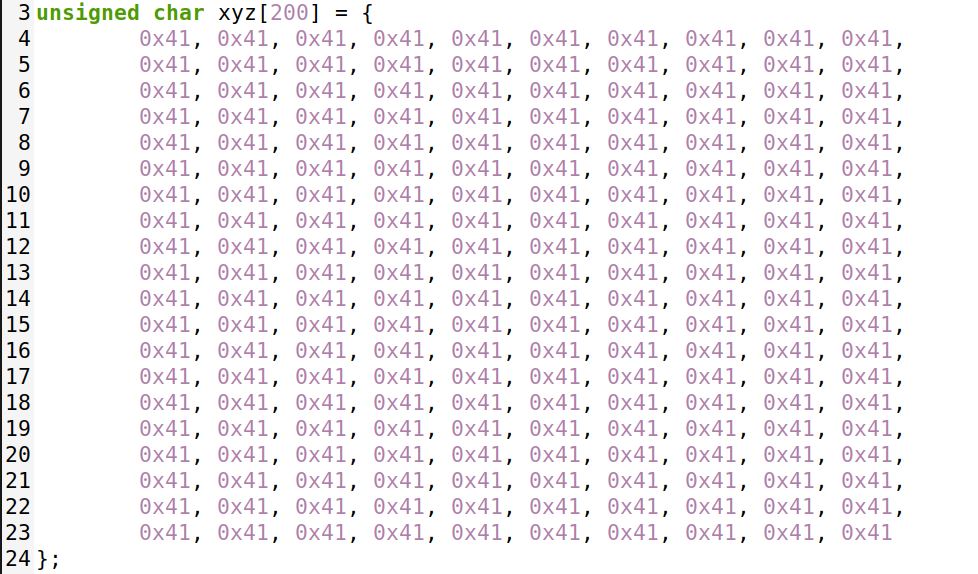


From the above experiment we can conclude that **the hash value doesn't change after the addition of a suffix and this property holds true for MD5.**

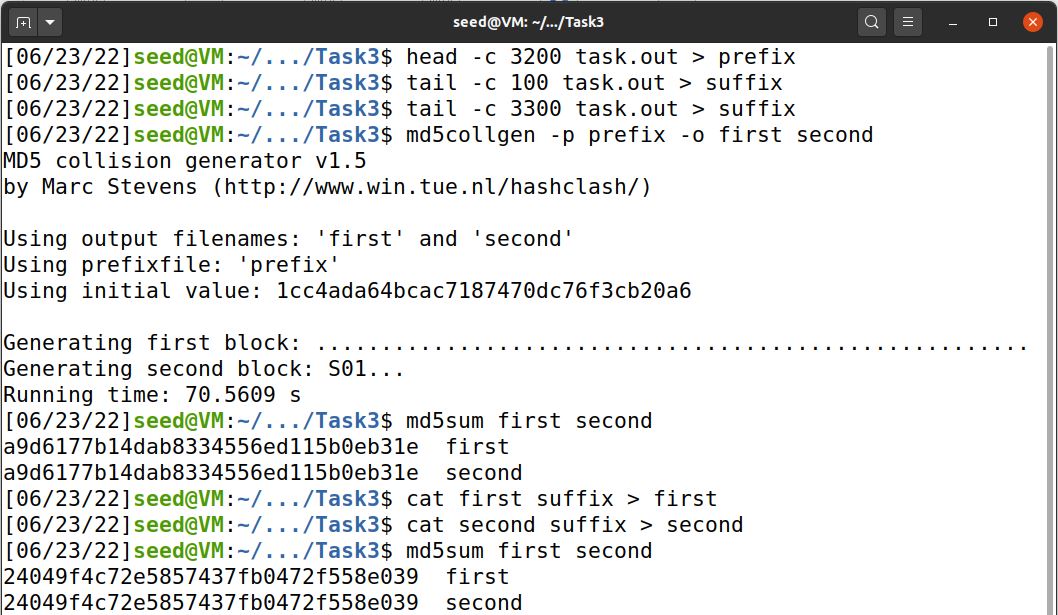
**Task 3: Generating Two Executable Files with the Same MD5 Hash**

Applying the aforementioned steps:

1. Initializing the values of the array
2. Generating the output in the file ‘Task.out’
3. Splitting the file ‘Task.out’ into ‘prefix’ and ‘suffix’
4. Generating two hash file from prefix, ‘first’ and ‘second’
5. Checking each of their values
6. Adding suffix in them
7. Checking their values



(step1)



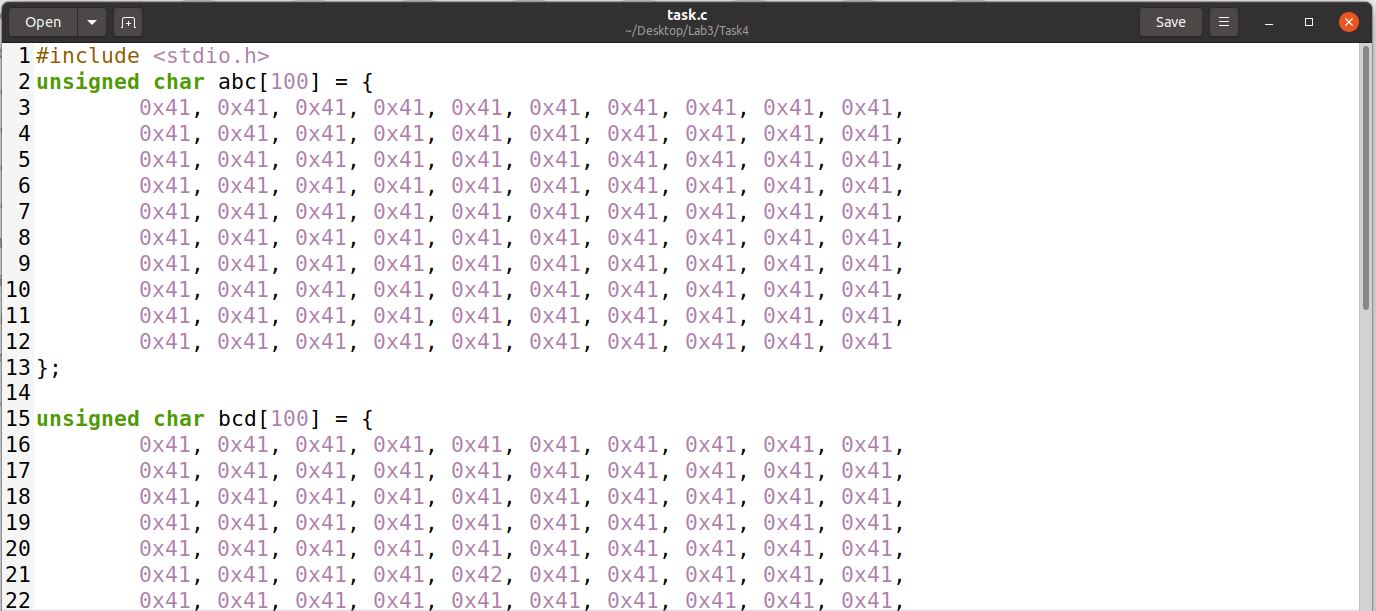
(step 2 to 7)



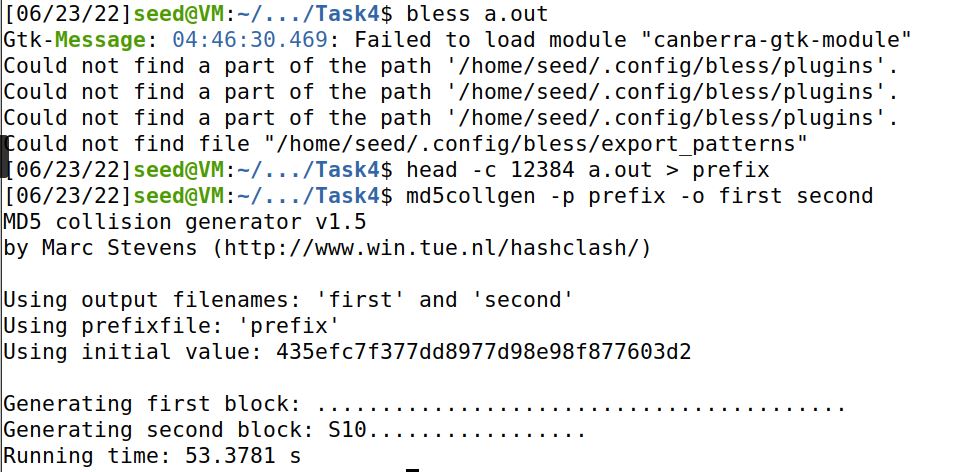
Generation of two files from one executable file does lead with the **same MD5 hash values with the presence of different prefix and suffix values.**

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

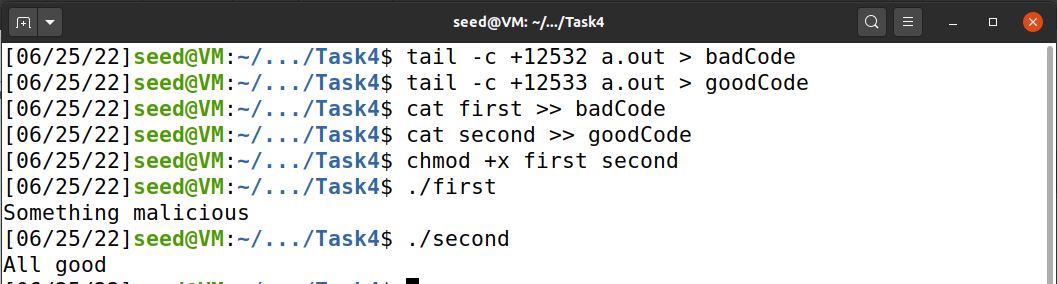
Making an executable C program:



Making Files with same MD5 hash codes, but resulting in different outcomes.



Combining them with different suffixes, and making them do malicious activities.



**Even though they have the MD5 hash values, simple difference in suffix makes them behave differently and possibly even maliciously.**