

Network Security

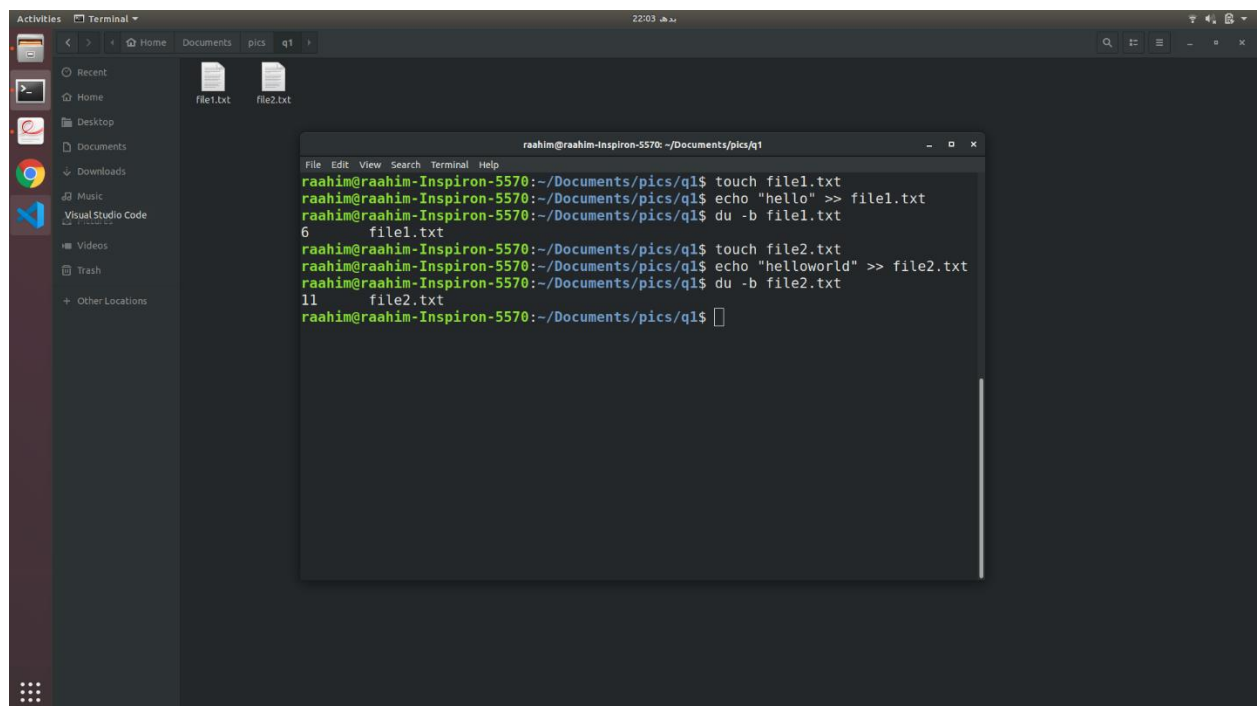
Cryptography Lab (One-Way Hash Functions and MAC)

Muhammad Raahim Khan

21100157

➤ Task1:

- 1) First I created two text files as shown in the screenshot below of sizes 6 and 11 bytes.

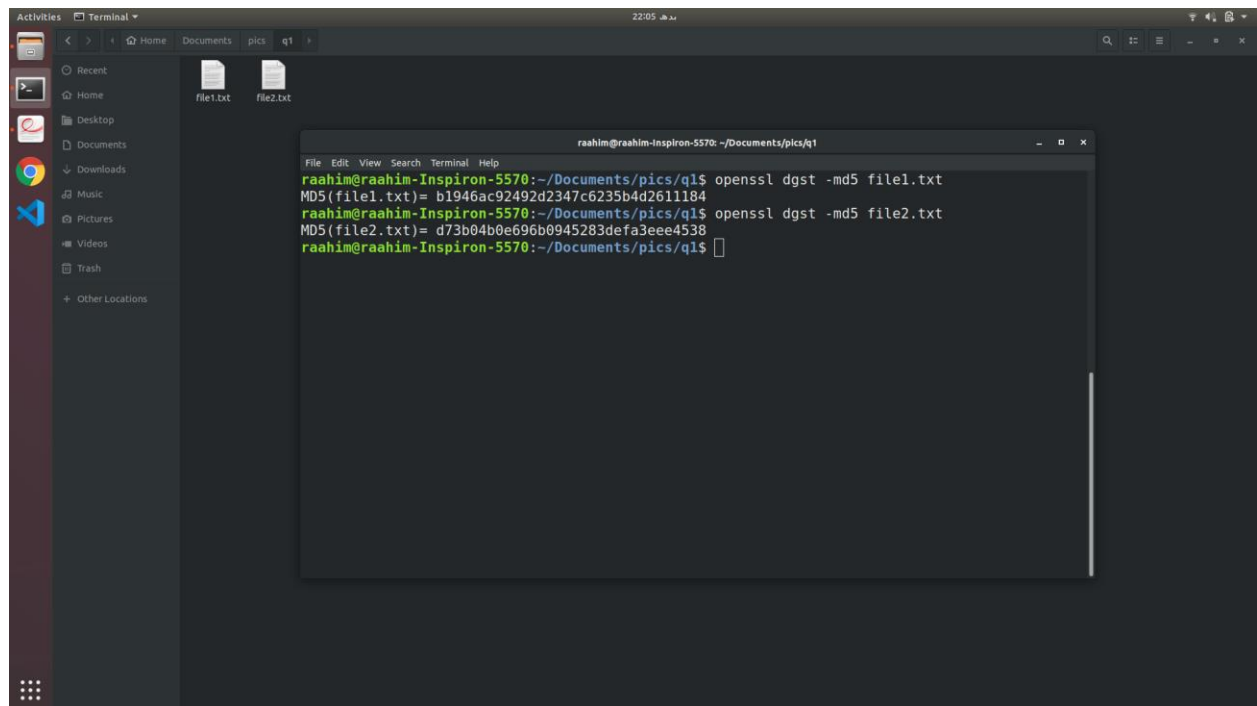


The screenshot shows a Linux desktop environment. In the background, a file manager window displays two files, 'file1.txt' and 'file2.txt', in the 'pics/q1' directory. In the foreground, a terminal window shows the following commands and output:

```
raahim@raahim-Inspiron-5570: ~/Documents/pics/q1
raahim@raahim-Inspiron-5570:~/Documents/pics/q1$ touch file1.txt
raahim@raahim-Inspiron-5570:~/Documents/pics/q1$ echo "hello" >> file1.txt
raahim@raahim-Inspiron-5570:~/Documents/pics/q1$ du -b file1.txt
6      file1.txt
raahim@raahim-Inspiron-5570:~/Documents/pics/q1$ touch file2.txt
raahim@raahim-Inspiron-5570:~/Documents/pics/q1$ echo "helloworld" >> file2.txt
raahim@raahim-Inspiron-5570:~/Documents/pics/q1$ du -b file2.txt
11     file2.txt
raahim@raahim-Inspiron-5570:~/Documents/pics/q1$
```

Afterwards, I computed the hash for both of these files using md5, sha1, and sha256 dgsttype.

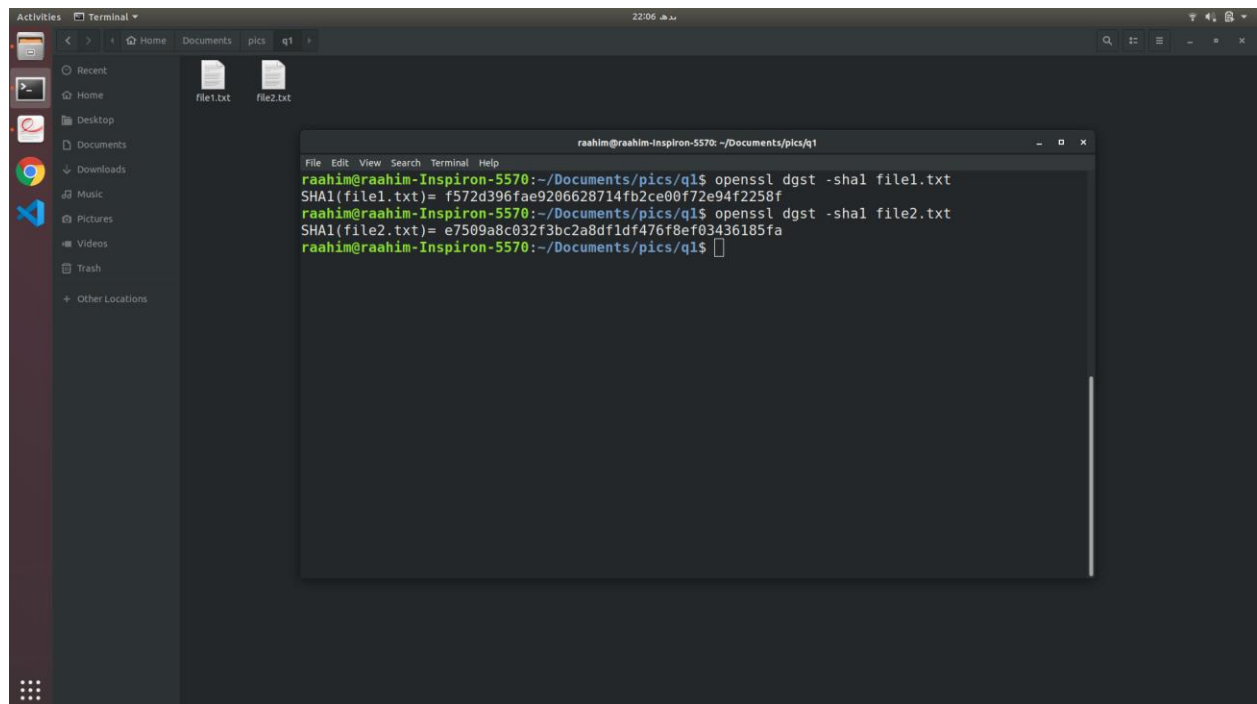
For md5:



The screenshot shows a Linux desktop environment with a terminal window open. The terminal is titled 'raahim@raahim-Inspiron-5570: ~/Documents/pics/q1'. The user has executed two commands to calculate MD5 hashes. The first command is 'openssl dgst -md5 file1.txt', which outputs 'MD5(file1.txt)= b1946ac92492d2347c6235b4d2611184'. The second command is 'openssl dgst -md5 file2.txt', which outputs 'MD5(file2.txt)= d73b04b0e696b0945283defa3eee4538'. The terminal window is overlaid on a file manager showing 'file1.txt' and 'file2.txt' in the 'pics' directory.

```
raahim@raahim-Inspiron-5570:~/Documents/pics/q1$ openssl dgst -md5 file1.txt
MD5(file1.txt)= b1946ac92492d2347c6235b4d2611184
raahim@raahim-Inspiron-5570:~/Documents/pics/q1$ openssl dgst -md5 file2.txt
MD5(file2.txt)= d73b04b0e696b0945283defa3eee4538
raahim@raahim-Inspiron-5570:~/Documents/pics/q1$
```

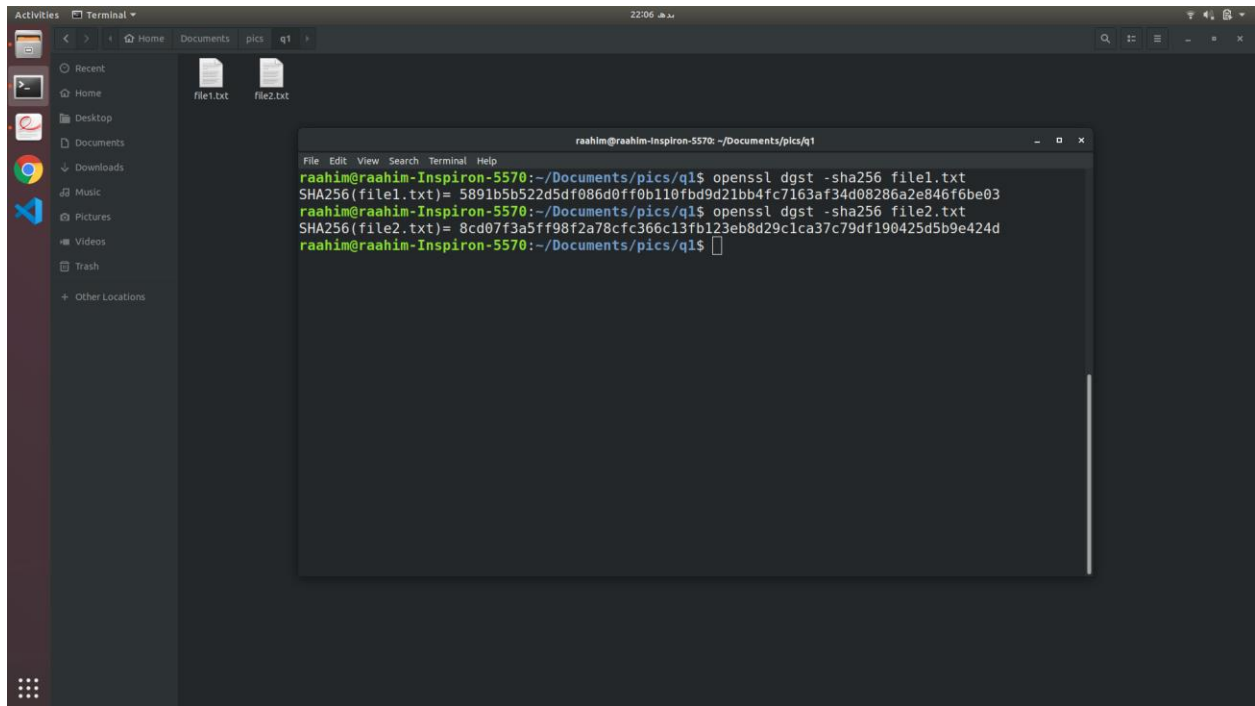
For sha1:



The screenshot shows the same Linux desktop environment as the previous one, but with a terminal window showing SHA1 hash calculations. The terminal is titled 'raahim@raahim-Inspiron-5570: ~/Documents/pics/q1'. The user has executed two commands to calculate SHA1 hashes. The first command is 'openssl dgst -sha1 file1.txt', which outputs 'SHA1(file1.txt)= f572d396fae9206628714fb2ce00f72e94f2258f'. The second command is 'openssl dgst -sha1 file2.txt', which outputs 'SHA1(file2.txt)= e7509a8c032f3bc2a8df1df476f8ef03436185fa'. The terminal window is overlaid on a file manager showing 'file1.txt' and 'file2.txt' in the 'pics' directory.

```
raahim@raahim-Inspiron-5570:~/Documents/pics/q1$ openssl dgst -sha1 file1.txt
SHA1(file1.txt)= f572d396fae9206628714fb2ce00f72e94f2258f
raahim@raahim-Inspiron-5570:~/Documents/pics/q1$ openssl dgst -sha1 file2.txt
SHA1(file2.txt)= e7509a8c032f3bc2a8df1df476f8ef03436185fa
raahim@raahim-Inspiron-5570:~/Documents/pics/q1$
```

For sha256:



The screenshot shows a Linux desktop environment with a terminal window open. The terminal displays the following commands and output:

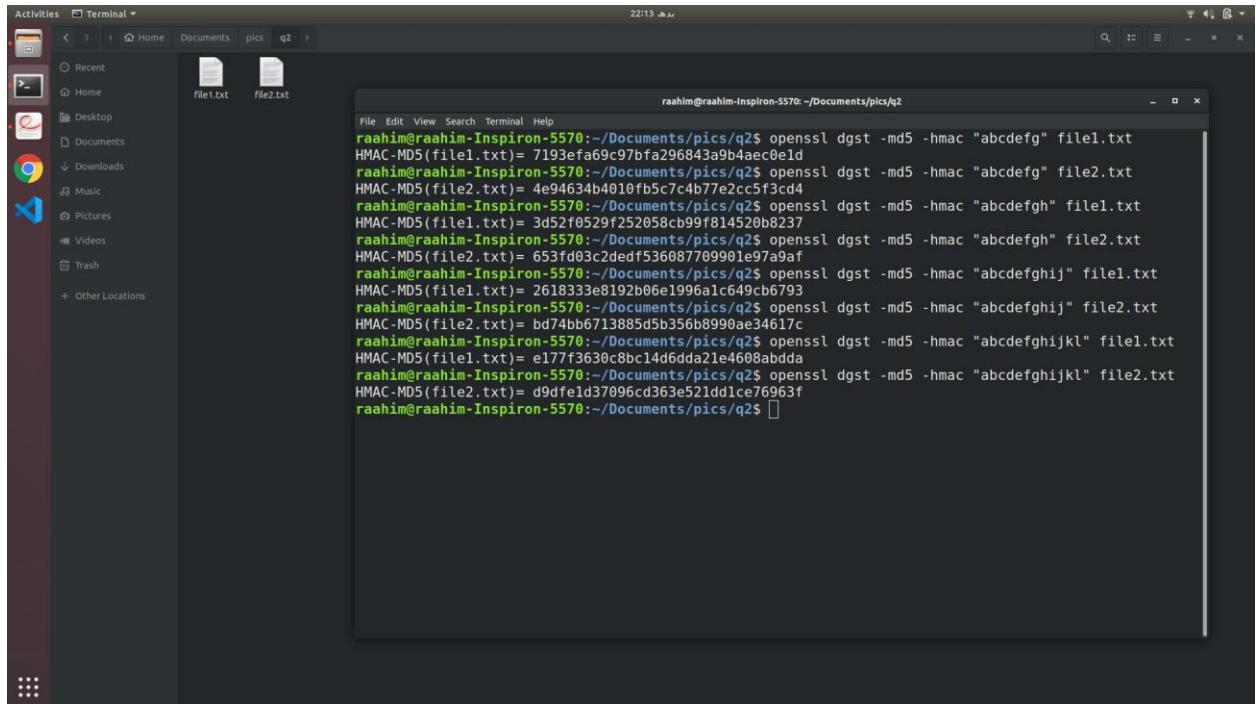
```
raahim@raahim-Inspiron-5570: ~/Documents/pics/q1
File Edit View Search Terminal Help
raahim@raahim-Inspiron-5570:~/Documents/pics/q1$ openssl dgst -sha256 file1.txt
SHA256(file1.txt)= 5891b5b522d5df086d0ff0b110fbd9d21bb4fc7163af34d08286a2e846f6be03
raahim@raahim-Inspiron-5570:~/Documents/pics/q1$ openssl dgst -sha256 file2.txt
SHA256(file2.txt)= 8cd07f3a5ff98f2a78cfc366c13fb123eb8d29c1ca37c79df190425d5b9e424d
raahim@raahim-Inspiron-5570:~/Documents/pics/q1$
```

From the above screenshots it can be seen that:

- For md5, hash computed is of 128 bits.
 - For sha1, hash computed is of 160 bits.
 - For sha256, hash computed is of 256 bits.
- 2) It can be seen from the screenshots above that digest size is not different for files of different lengths. Hashes were generated using files of different lengths. The resulting hashes were all of similar length (if same category of hash is used on files of different lengths).
-

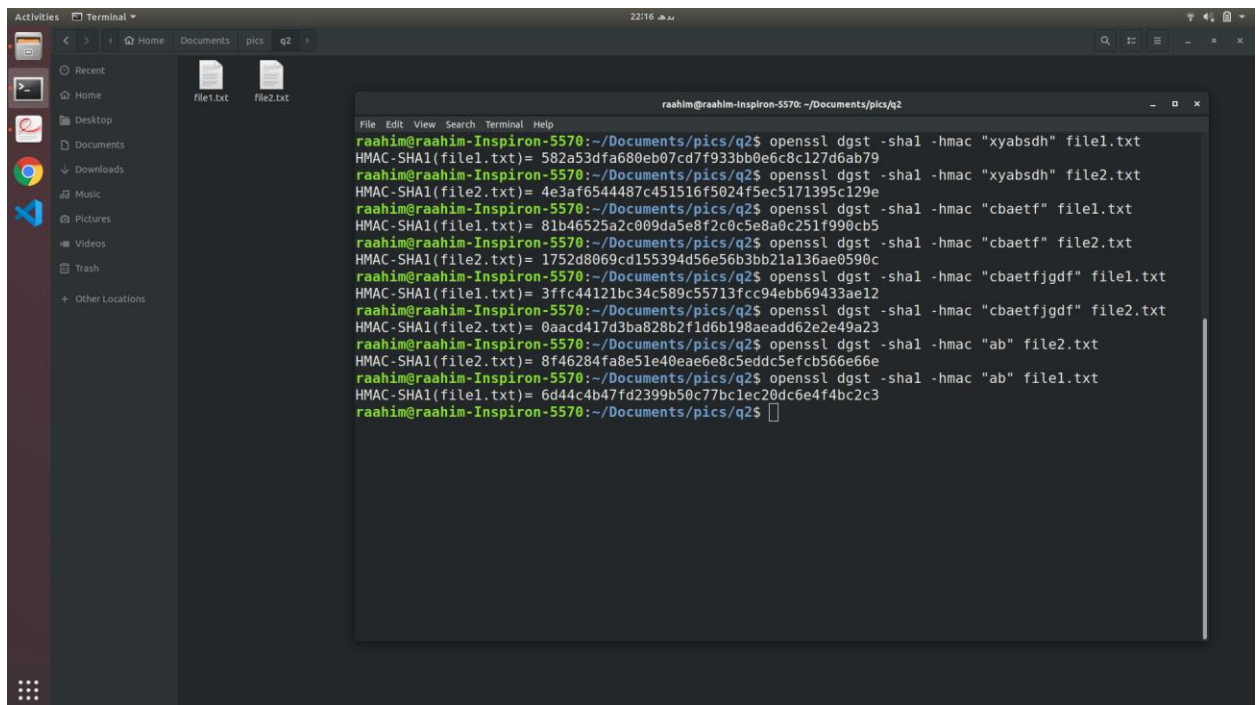
➤ Task2:

Keyed hash generated using HMAC-MD5:



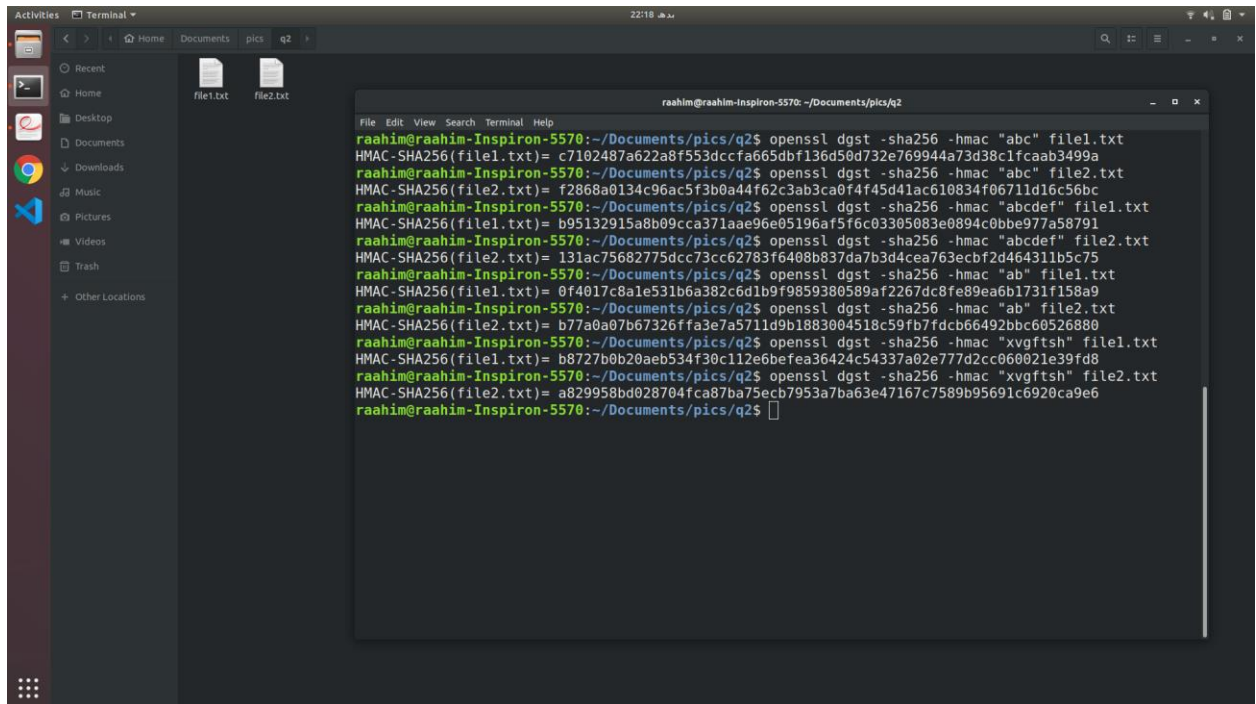
```
raahim@raahim-Inspiron-5570: ~/Documents/pics/q2
raahim@raahim-Inspiron-5570:~/Documents/pics/q2$ openssl dgst -md5 -hmac "abcdefg" file1.txt
HMAC-MD5(file1.txt)= 7193efa69c97bfa296843a9b4aec0e1d
raahim@raahim-Inspiron-5570:~/Documents/pics/q2$ openssl dgst -md5 -hmac "abcdefg" file2.txt
HMAC-MD5(file2.txt)= 4e94634b4810fb5c7c4b77e2cc5f3cd4
raahim@raahim-Inspiron-5570:~/Documents/pics/q2$ openssl dgst -md5 -hmac "abcdefgh" file1.txt
HMAC-MD5(file1.txt)= 3d52f0529f252058cb99f814520b8237
raahim@raahim-Inspiron-5570:~/Documents/pics/q2$ openssl dgst -md5 -hmac "abcdefgh" file2.txt
HMAC-MD5(file2.txt)= 653fd03c2dedf536087709901e97a9af
raahim@raahim-Inspiron-5570:~/Documents/pics/q2$ openssl dgst -md5 -hmac "abcdefghij" file1.txt
HMAC-MD5(file1.txt)= 2618333e8192b06e1996a1c649cb6793
raahim@raahim-Inspiron-5570:~/Documents/pics/q2$ openssl dgst -md5 -hmac "abcdefghij" file2.txt
HMAC-MD5(file2.txt)= bd74bb6713885d5b356b8990ae34617c
raahim@raahim-Inspiron-5570:~/Documents/pics/q2$ openssl dgst -md5 -hmac "abcdefghijkl" file1.txt
HMAC-MD5(file1.txt)= e177f3630c8bc14d6dda21e4608abdda
raahim@raahim-Inspiron-5570:~/Documents/pics/q2$ openssl dgst -md5 -hmac "abcdefghijkl" file2.txt
HMAC-MD5(file2.txt)= d9dfeld37096cd363e521dd1ce76963f
raahim@raahim-Inspiron-5570:~/Documents/pics/q2$
```

Keyed hash generated using HMAC-SHA1:



```
raahim@raahim-Inspiron-5570: ~/Documents/pics/q2
raahim@raahim-Inspiron-5570:~/Documents/pics/q2$ openssl dgst -sha1 -hmac "xyabsdh" file1.txt
HMAC-SHA1(file1.txt)= 582a53dfa680eb07cd7f933bb0e6c8c127d6ab79
raahim@raahim-Inspiron-5570:~/Documents/pics/q2$ openssl dgst -sha1 -hmac "xyabsdh" file2.txt
HMAC-SHA1(file2.txt)= 4e3af6544487c451516f5024f5ec5171395c129e
raahim@raahim-Inspiron-5570:~/Documents/pics/q2$ openssl dgst -sha1 -hmac "cbaetf" file1.txt
HMAC-SHA1(file1.txt)= 81b46525a2c009da5e8f2c0c5e8a0c251f990cb5
raahim@raahim-Inspiron-5570:~/Documents/pics/q2$ openssl dgst -sha1 -hmac "cbaetf" file2.txt
HMAC-SHA1(file2.txt)= 1752d8069cd155394d56e56b3bb21a136ae0590c
raahim@raahim-Inspiron-5570:~/Documents/pics/q2$ openssl dgst -sha1 -hmac "cbaetfjgdf" file1.txt
HMAC-SHA1(file1.txt)= 3ffc44121bc34c589c55713fcc94ebb69433ae12
raahim@raahim-Inspiron-5570:~/Documents/pics/q2$ openssl dgst -sha1 -hmac "cbaetfjgdf" file2.txt
HMAC-SHA1(file2.txt)= 0aacd417d3ba828b2f1d6b198aeadd62e2e49a23
raahim@raahim-Inspiron-5570:~/Documents/pics/q2$ openssl dgst -sha1 -hmac "ab" file2.txt
HMAC-SHA1(file2.txt)= 8f46284fa8e51e40eae6e8c5eddc5efcb566e66e
raahim@raahim-Inspiron-5570:~/Documents/pics/q2$ openssl dgst -sha1 -hmac "ab" file1.txt
HMAC-SHA1(file1.txt)= 6d44c4b47fd2399b50c77bclcc20dc6e4f4bc2c3
raahim@raahim-Inspiron-5570:~/Documents/pics/q2$
```

Keyed hash generated using HMAC-SHA256:



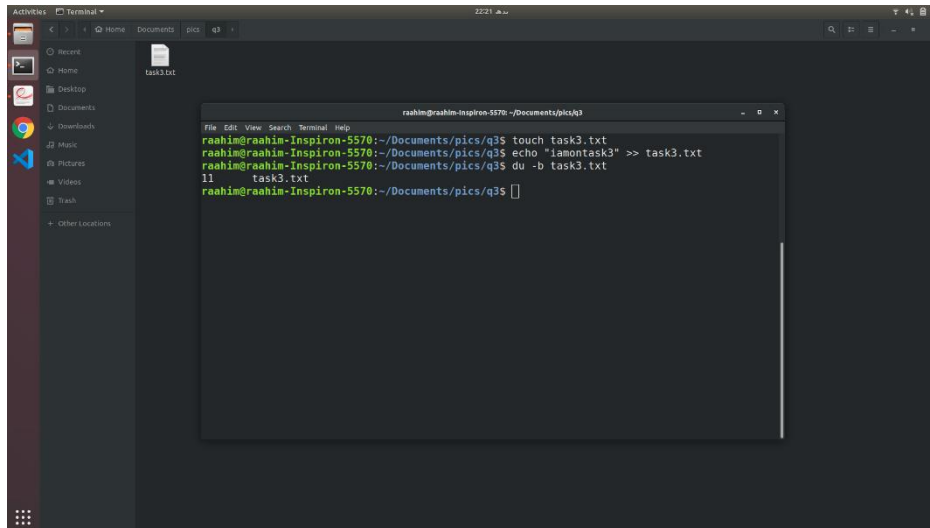
The screenshot shows a terminal window with the following commands and output:

```
raahim@raahim-Inspiron-5570: ~/Documents/pics/q2
raahim@raahim-Inspiron-5570:~/Documents/pics/q2$ openssl dgst -sha256 -hmac "abc" file1.txt
HMAC-SHA256(file1.txt)= c7102487a622a8f553dcca665dbf136d50d732e769944a73d38c1fcaab3499a
raahim@raahim-Inspiron-5570:~/Documents/pics/q2$ openssl dgst -sha256 -hmac "abc" file2.txt
HMAC-SHA256(file2.txt)= f2868a0134c96ac5f3b0a44f62c3ab3ca0f4f45d41ac610834f06711d16c56bc
raahim@raahim-Inspiron-5570:~/Documents/pics/q2$ openssl dgst -sha256 -hmac "abcdef" file1.txt
HMAC-SHA256(file1.txt)= b95132915a8b09cca371aae96e05196af5f6c03305083e0894c0bbe977a58791
raahim@raahim-Inspiron-5570:~/Documents/pics/q2$ openssl dgst -sha256 -hmac "abcdef" file2.txt
HMAC-SHA256(file2.txt)= 131ac75682775dcc73cc62783f6408b837da7b3d4cea763ecbf2d464311b5c75
raahim@raahim-Inspiron-5570:~/Documents/pics/q2$ openssl dgst -sha256 -hmac "ab" file1.txt
HMAC-SHA256(file1.txt)= 0f4017c8a1e531b6a382c6d1b9f9859380589af2267dc8fe89ea6b1731f158a9
raahim@raahim-Inspiron-5570:~/Documents/pics/q2$ openssl dgst -sha256 -hmac "ab" file2.txt
HMAC-SHA256(file2.txt)= b77a0a07b67326ffa3e7a5711d9b1883004518c59fb7fdbc66492bbc60526880
raahim@raahim-Inspiron-5570:~/Documents/pics/q2$ openssl dgst -sha256 -hmac "xvgftsh" file1.txt
HMAC-SHA256(file1.txt)= b8727b0b20aeb534f30c112e6befea36424c54337a02e77742cc060021e39fd8
raahim@raahim-Inspiron-5570:~/Documents/pics/q2$ openssl dgst -sha256 -hmac "xvgftsh" file2.txt
HMAC-SHA256(file2.txt)= a829958bd028704fca87ba75ecb7953a7ba63e47167c7589b95691c6920ca9e6
raahim@raahim-Inspiron-5570:~/Documents/pics/q2$
```

- 1) No, we do not have to use a key with a fixed size in HMAC.
- 2) We do not have to use a key with a fixed size in HMAC, hence, there is no fixed key size.
- 3) Reason is “no” because HMAC is a cryptographic hash function which map data of arbitrary size to a bit string of a fixed size (a hash). This is opposite to block ciphers which typically need a fixed length key. According to <https://tools.ietf.org/html/rfc2104> “The key for HMAC can be of any length...”.

➤ Task3:

First, a text file of size 11 bytes was created (two times: first for MD5 and then for SHA256)

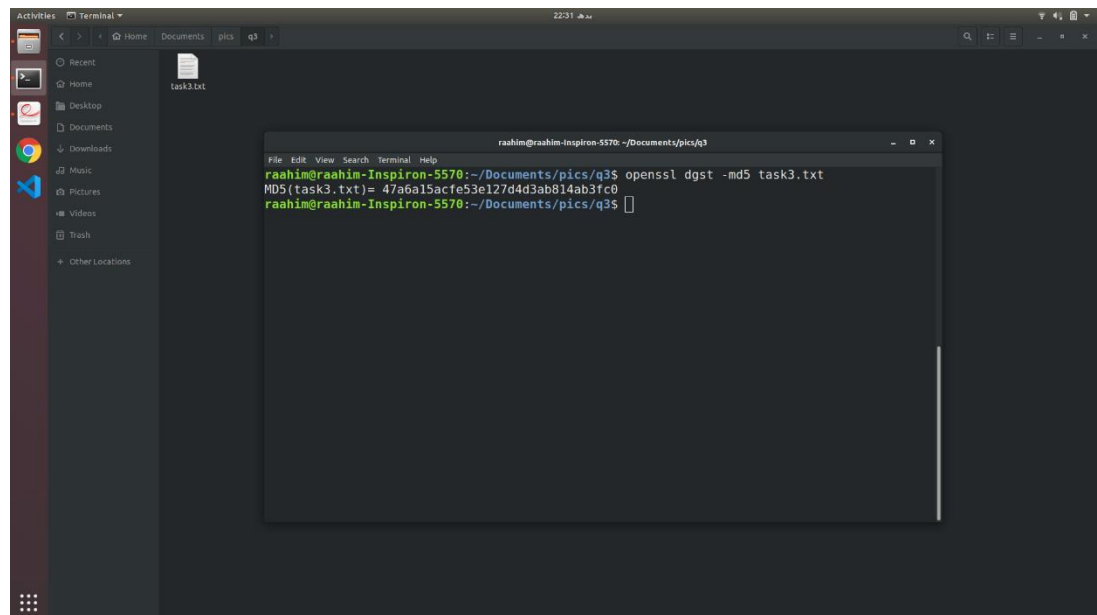
A terminal window titled 'raahim@raahim-inspiron-5570: ~/Documents/pics/q3' is shown. The terminal output displays the following commands and results:

```
raahim@raahim-inspiron-5570:~/Documents/pics/q3$ touch task3.txt
raahim@raahim-inspiron-5570:~/Documents/pics/q3$ echo "iamontask3" >> task3.txt
raahim@raahim-inspiron-5570:~/Documents/pics/q3$ du -b task3.txt
11 task3.txt
raahim@raahim-inspiron-5570:~/Documents/pics/q3$
```

The file manager on the left shows a file named 'task3.txt' in the 'q3' directory.

- For MD5:

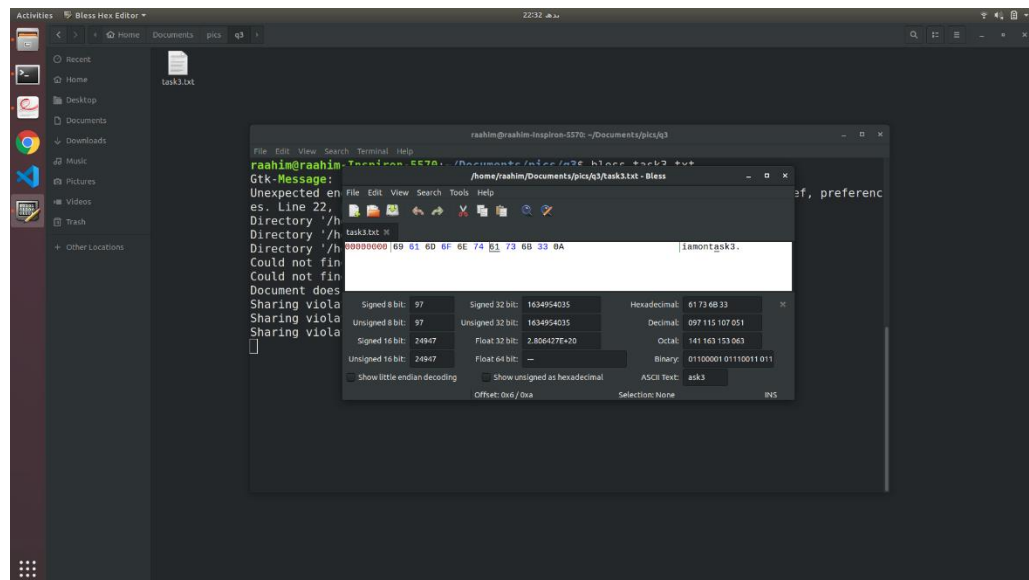
H1 was as follows:

A terminal window titled 'raahim@raahim-inspiron-5570: ~/Documents/pics/q3' is shown. The terminal output displays the following commands and results:

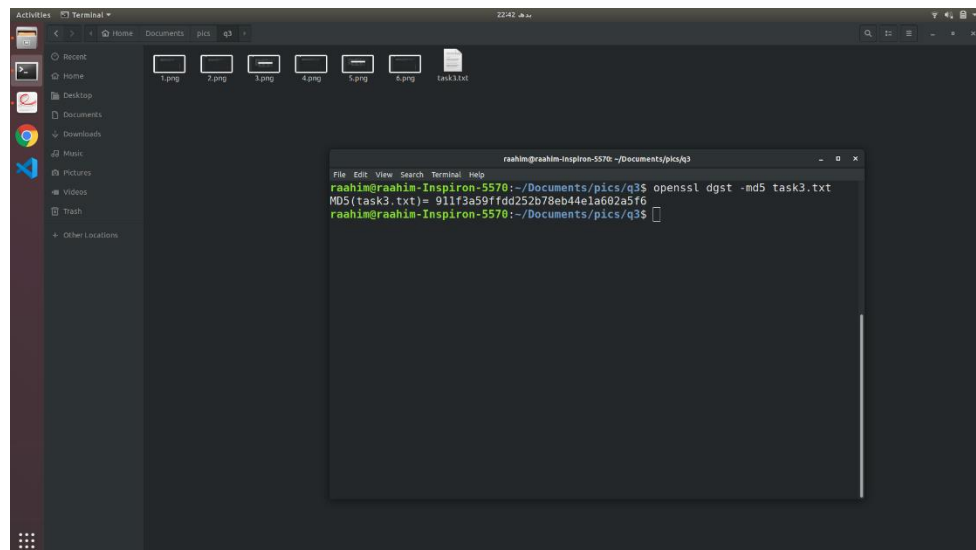
```
raahim@raahim-inspiron-5570:~/Documents/pics/q3$ openssl dgst -md5 task3.txt
MD5(task3.txt)= 47a6a15acfe53e127d4d3ab814ab3fc0
raahim@raahim-inspiron-5570:~/Documents/pics/q3$
```

The file manager on the left shows a file named 'task3.txt' in the 'q3' directory.

Using Bless, the underlined byte was changed from 61 to 67 i.e. 1 was changed to 7:

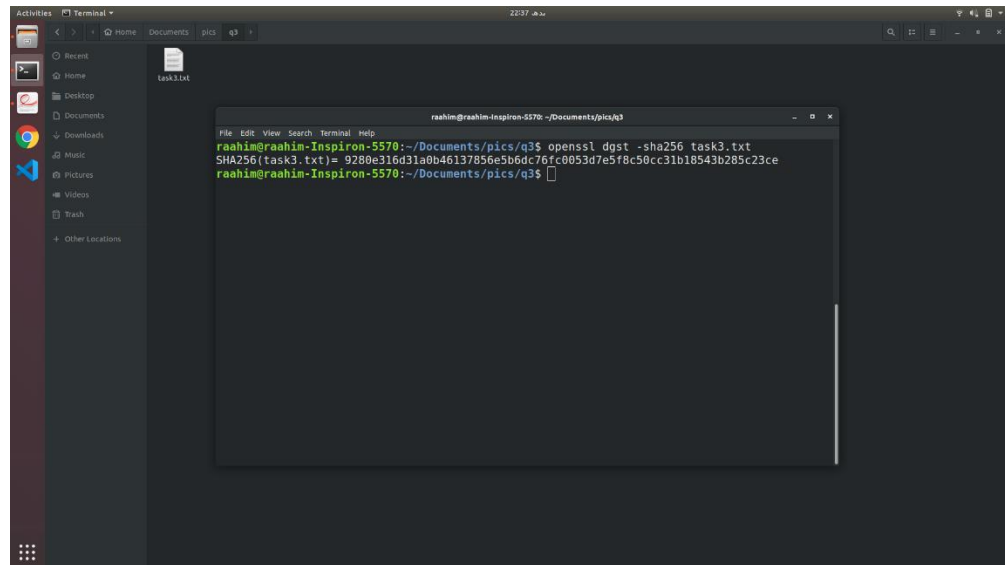


H2 was as follows:



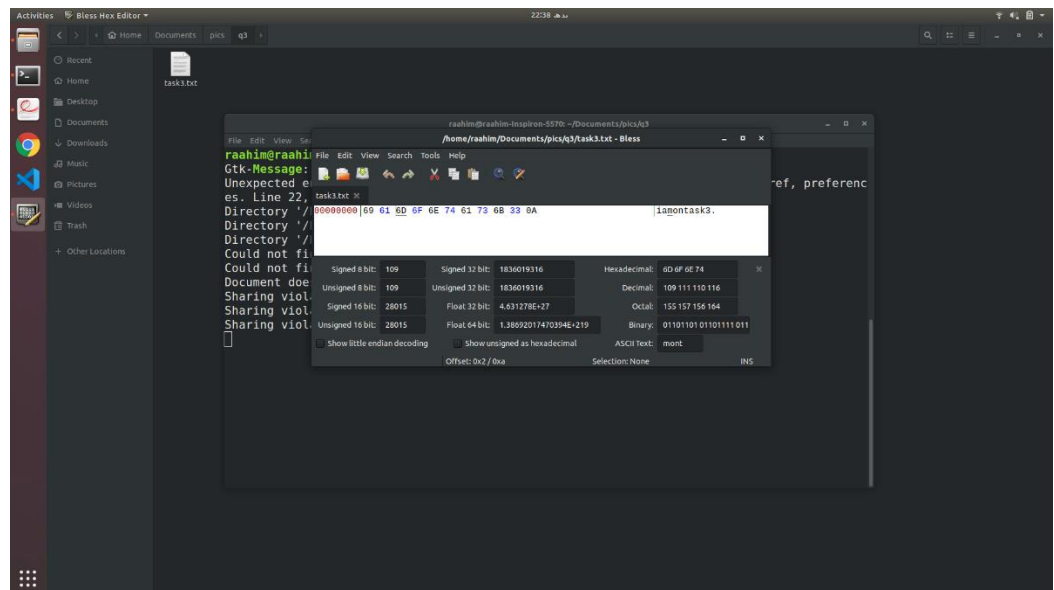
- For SHA256:

H1 was as follows:



A terminal window titled 'raahim@raahim-Inspiron-5570: ~/Documents/pics/q3' shows the command `openssl dgst -sha256 task3.txt` being executed. The output is `SHA256(task3.txt)= 9280e316d31a0b46137856e5b6dc76fc0053d7e5f8c50cc31b18543b285c23ce`. The file 'task3.txt' is visible in the background file manager.

Using Bless, the underlined byte was changed from 6D to 6A i.e. D was changed to A:

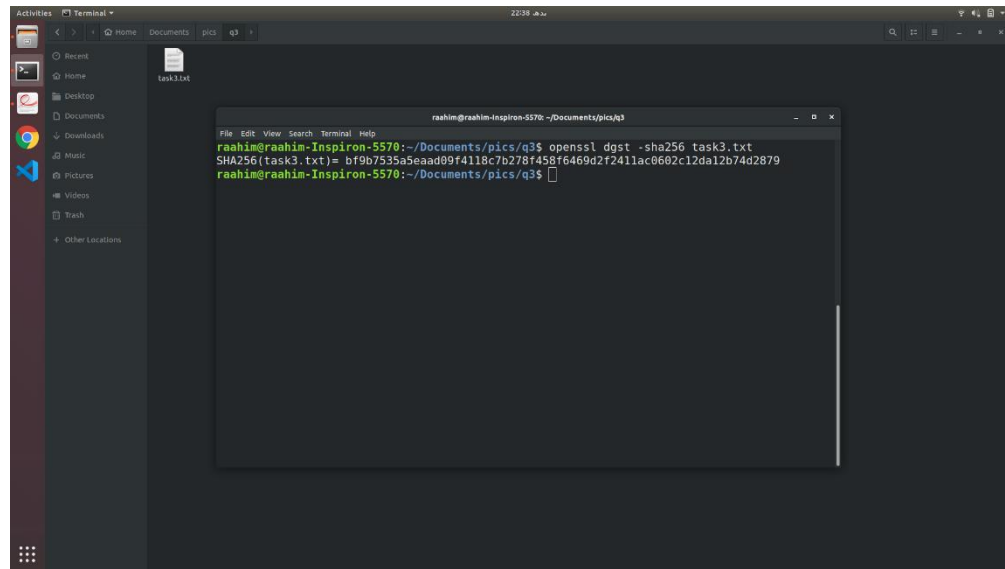


A Bless Hex Editor window titled 'raahim@raahim-Inspiron-5570: ~/Documents/pics/q3' is shown editing 'task3.txt'. The hex editor displays a hex dump where the byte '6D' at offset 00000000 is highlighted. A 'Gtk+Message' dialog box is open, showing a warning about an 'Unexpected error' and a 'Directory' path. The dialog also contains a table of conversion data:

Signed 8 bit:	Unsigned 8 bit:	Signed 32 bit:	Unsigned 32 bit:	Hexadecimal:	Decimal:
109	109	1836019316	1836019316	6D 6F 0E 74	109 111 110 116
					Octal: 155 157 156 164
					Binary: 01101101 01101111 0111

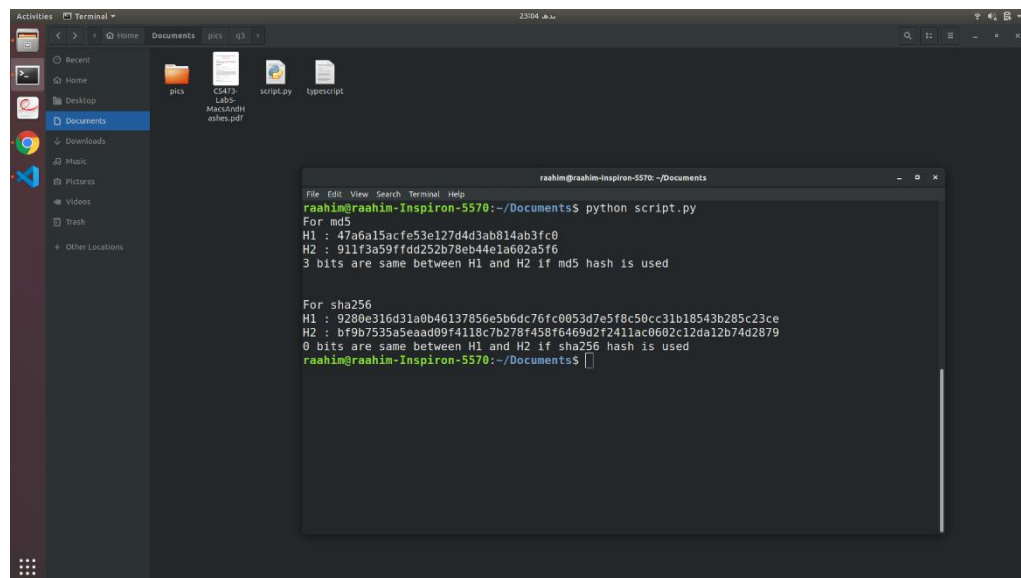
The dialog also includes checkboxes for 'Show little endian decoding', 'Show unsigned as hexadecimal', and 'Offset: 0x2 / 0xa', and a 'Selection: None' field.

H2 was as follows:

A terminal window titled 'raahim@raahim-Inspiron-5570: ~/Documents/pics/q3' is open. The user has executed the command 'openssl dgst -sha256 task3.txt'. The output shows the SHA256 hash of 'task3.txt' as 'bf9b7535a5eaad09f4118c7b278f458f6469d2f2411ac0602c12da12b74d2879'.

```
raahim@raahim-Inspiron-5570:~/Documents/pics/q3$ openssl dgst -sha256 task3.txt
SHA256(task3.txt)= bf9b7535a5eaad09f4118c7b278f458f6469d2f2411ac0602c12da12b74d2879
raahim@raahim-Inspiron-5570:~/Documents/pics/q3$
```

I wrote a short script to count how many bits are the same between H1 and H2 and the results were as follows:

A terminal window titled 'raahim@raahim-Inspiron-5570: ~/Documents' is open. The user has executed the command 'python script.py'. The script outputs the MD5 and SHA256 hashes for H1 and H2, and then reports the number of bits that are the same for each algorithm.

```
raahim@raahim-Inspiron-5570:~/Documents$ python script.py
For md5
H1 : 47a6a15acfe53e127d4d3ab814ab3fc0
H2 : 911f3a59ffdd252b78eb44e1a602a5f6
3 bits are same between H1 and H2 if md5 hash is used

For sha256
H1 : 9200e316d31a0b46137856e5b6dc76fc0853d7e5f8c50ce31b18543b285c23ce
H2 : bf9b7535a5eaad09f4118c7b278f458f6469d2f2411ac0602c12da12b74d2879
0 bits are same between H1 and H2 if sha256 hash is used
raahim@raahim-Inspiron-5570:~/Documents$
```

It can be seen that 3 bits are same between H1 and H2 using MD5 algorithm whereas using SHA256 algorithm, there is no same bit between H1 and H2.
