

Tarea # 6 Firma

1.- En nuestra línea de comandos verificamos nuestra ruta y generamos un archivo.c con lo siguiente:

```
MINGW64:/c/Users/PC
PC@DESKTOP-5II5100 MINGW64 ~
$ pwd
/c/Users/PC
PC@DESKTOP-5II5100 MINGW64 ~
$ nano hola.c
PC@DESKTOP-5II5100 MINGW64 ~
$ cat hola.c
/* Hola AMIGOS DE YOUTUBE!! */
/* Probando la Firmaa */

#include <conio.h>
#include <stdio.h>

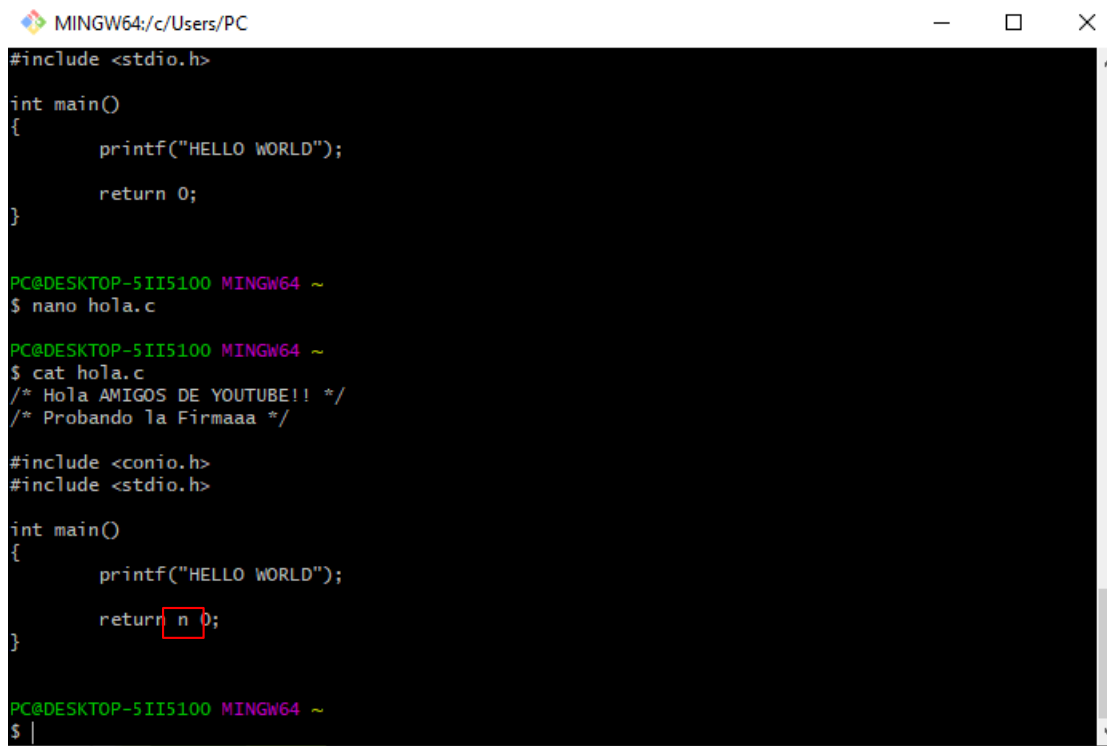
int main()
{
    printf("HELLO WORLD");

    return 0;
}
PC@DESKTOP-5II5100 MINGW64 ~
$ |
```

2.- Generamos el hash de nuestro archivo y guardamos en un archivo aparte firma.txt:

```
MINGW64:/c/Users/PC
quartus_full_rules_file.txt
quartus_web_rules_file.txt
quartus2.ini
quartus2.qreg
Reciente@
saludo.txt.sha256
'Saved Games'/
School-ERP-Intro/
Searches/
SendTo@
source/
Sti_Trace.log
Tetris_full.py
Tetriscancion.mp3
Twitch/
Videos/
PC@DESKTOP-5II5100 MINGW64 ~
$ pwd
/c/Users/PC
PC@DESKTOP-5II5100 MINGW64 ~
$ shasum -a 1 hola.c
24ec4c7810126d31e445c0b863ee9bd71c575740 *hola.c
PC@DESKTOP-5II5100 MINGW64 ~
$ cat firma.txt
hola.c 1
24ec4c7810126d31e445c0b863ee9bd71c575740
```

3.- Realizamos un cambio marcado con un cuadro rojo dentro de nuestro archivo.c:



```
MINGW64:/c/Users/PC
#include <stdio.h>

int main()
{
    printf("HELLO WORLD");

    return 0;
}

PC@DESKTOP-5II5100 MINGW64 ~
$ nano hola.c

PC@DESKTOP-5II5100 MINGW64 ~
$ cat hola.c
/* Hola AMIGOS DE YOUTUBE!! */
/* Probando la Firmaaa */

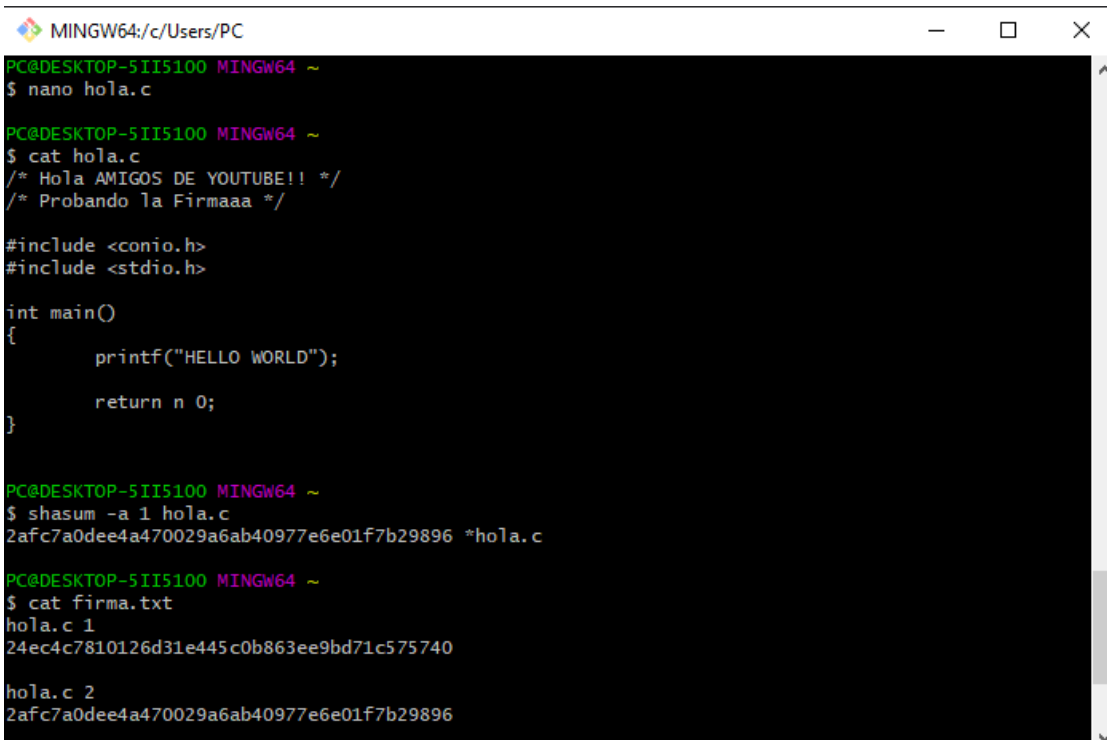
#include <conio.h>
#include <stdio.h>

int main()
{
    printf("HELLO WORLD");

    return n 0;
}

PC@DESKTOP-5II5100 MINGW64 ~
$ |
```

4.- Notamos el cambio de nuestro hash generado y volvemos a guardar en él .txt:



```
MINGW64:/c/Users/PC
PC@DESKTOP-5II5100 MINGW64 ~
$ nano hola.c

PC@DESKTOP-5II5100 MINGW64 ~
$ cat hola.c
/* Hola AMIGOS DE YOUTUBE!! */
/* Probando la Firmaaa */

#include <conio.h>
#include <stdio.h>

int main()
{
    printf("HELLO WORLD");

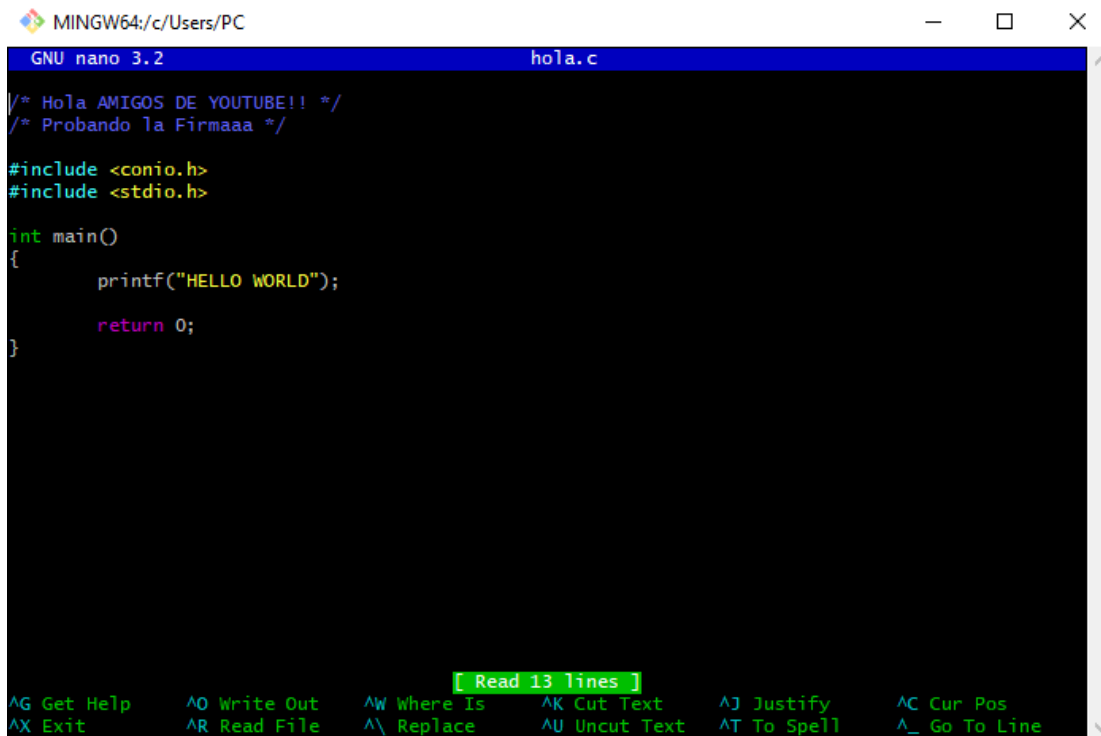
    return n 0;
}

PC@DESKTOP-5II5100 MINGW64 ~
$ shasum -a 1 hola.c
2afc7a0dee4a470029a6ab40977e6e01f7b29896 *hola.c

PC@DESKTOP-5II5100 MINGW64 ~
$ cat firma.txt
hola.c 1
24ec4c7810126d31e445c0b863ee9bd71c575740

hola.c 2
2afc7a0dee4a470029a6ab40977e6e01f7b29896
```

5.- Regresamos al estado original el archivo .c para observar que genera el comando hash:



The screenshot shows a terminal window titled 'MINGW64:/c/Users/PC' with a nano 3.2 editor open to a file named 'hola.c'. The code is as follows:

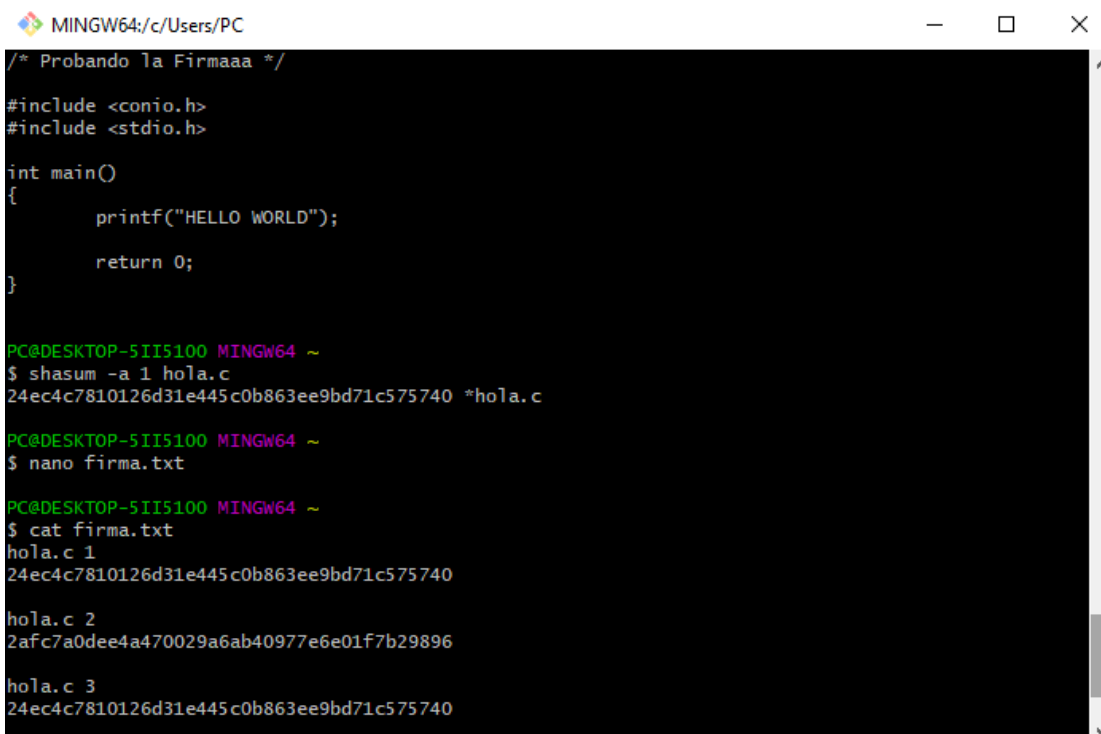
```
/* Hola AMIGOS DE YOUTUBE!! */
/* Probando la Firmaa */

#include <conio.h>
#include <stdio.h>

int main()
{
    printf("HELLO WORLD");
    return 0;
}
```

The status bar at the bottom indicates '[Read 13 lines]' and lists various keyboard shortcuts for nano.

6.- Efectivamente vuelve a generar el primer hash como si nada hubiera ocurrido y la evidencia está:



The screenshot shows a terminal window titled 'MINGW64:/c/Users/PC' with the following commands and output:

```
/* Probando la Firmaa */

#include <conio.h>
#include <stdio.h>

int main()
{
    printf("HELLO WORLD");
    return 0;
}

PC@DESKTOP-5II5100 MINGW64 ~
$ shasum -a 1 hola.c
24ec4c7810126d31e445c0b863ee9bd71c575740 *hola.c

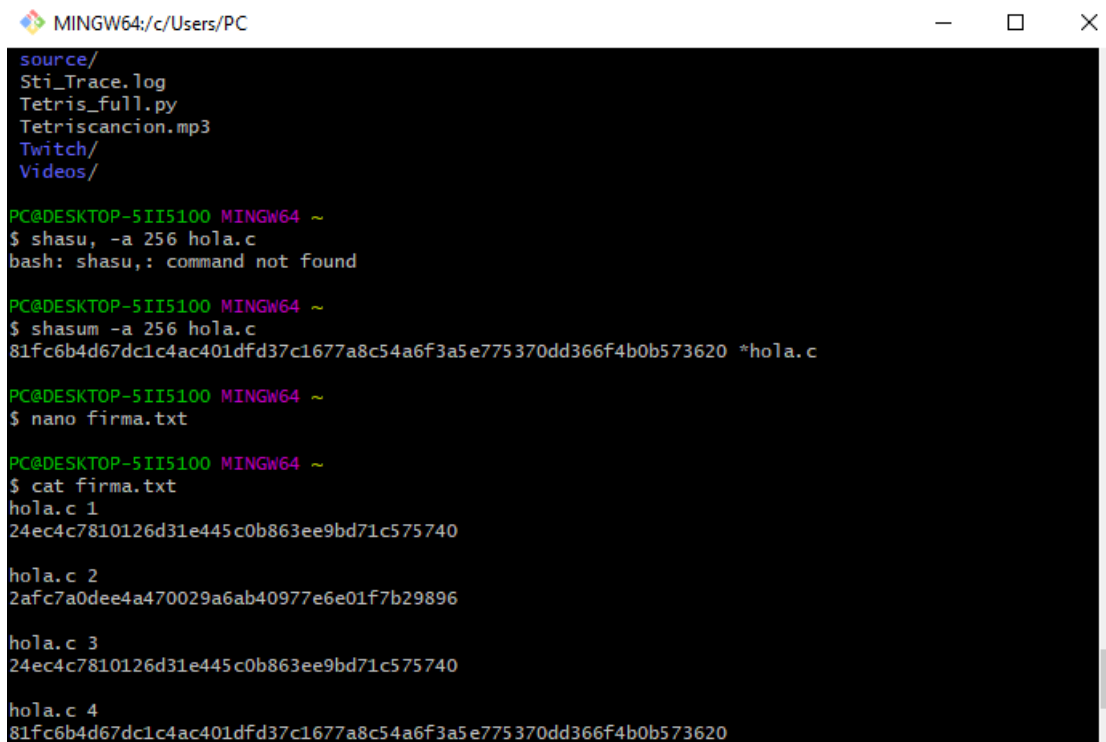
PC@DESKTOP-5II5100 MINGW64 ~
$ nano firma.txt

PC@DESKTOP-5II5100 MINGW64 ~
$ cat firma.txt
hola.c 1
24ec4c7810126d31e445c0b863ee9bd71c575740

hola.c 2
2afc7a0dee4a470029a6ab40977e6e01f7b29896

hola.c 3
24ec4c7810126d31e445c0b863ee9bd71c575740
```

7.- Generamos el hash con el algoritmo 256 para observar la diferencia y guardamos en .txt:



```
MINGW64:/c/Users/PC
source/
Sti_Trace.log
Tetris_full.py
Tetriscancion.mp3
Twitch/
Videos/

PC@DESKTOP-5II5100 MINGW64 ~
$ shasu, -a 256 hola.c
bash: shasu,: command not found

PC@DESKTOP-5II5100 MINGW64 ~
$ shasum -a 256 hola.c
81fc6b4d67dc1c4ac401dfd37c1677a8c54a6f3a5e775370dd366f4b0b573620 *hola.c

PC@DESKTOP-5II5100 MINGW64 ~
$ nano firma.txt

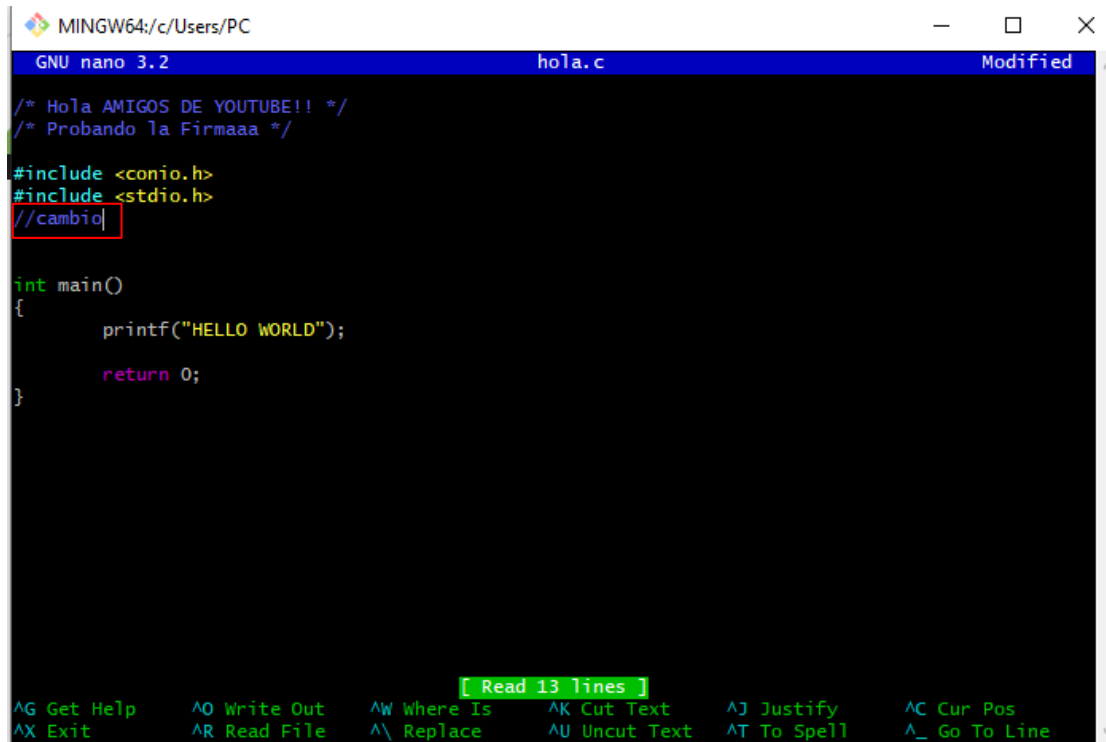
PC@DESKTOP-5II5100 MINGW64 ~
$ cat firma.txt
hola.c 1
24ec4c7810126d31e445c0b863ee9bd71c575740

hola.c 2
2afc7a0dee4a470029a6ab40977e6e01f7b29896

hola.c 3
24ec4c7810126d31e445c0b863ee9bd71c575740

hola.c 4
81fc6b4d67dc1c4ac401dfd37c1677a8c54a6f3a5e775370dd366f4b0b573620
```

8.- Evidencia del “//cambio” realizado en el archivo para observar nuevamente el ultimo hash:



```
GNU nano 3.2 hola.c Modified
/* Hola AMIGOS DE YOUTUBE!! */
/* Probando la Firmaaa */

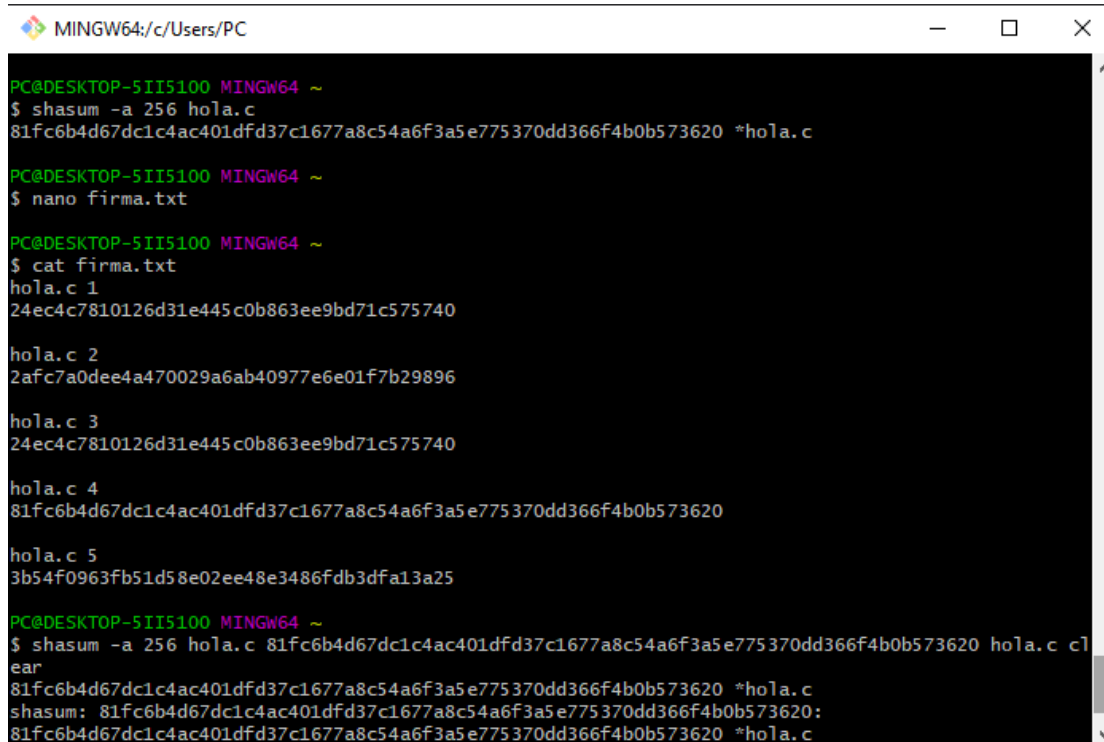
#include <conio.h>
#include <stdio.h>
//cambio

int main()
{
    printf("HELLO WORLD");

    return 0;
}

[ Read 13 lines ]
AG Get Help  AO Write Out  AW Where Is  AK Cut Text  AJ Justify  AC Cur Pos
AX Exit      AR Read File  A\ Replace  AU Uncut Text AT To Spell  ^_ Go To Line
```

9.- Todos los hash generados guardados en nuestro archivo .txt y además una comprobación extra:



```
PC@DESKTOP-5II5100 MINGW64 ~  
$ shasum -a 256 ho!a.c  
81fc6b4d67dc1c4ac401dfd37c1677a8c54a6f3a5e775370dd366f4b0b573620 *ho!a.c  
  
PC@DESKTOP-5II5100 MINGW64 ~  
$ nano firma.txt  
  
PC@DESKTOP-5II5100 MINGW64 ~  
$ cat firma.txt  
ho!a.c 1  
24ec4c7810126d31e445c0b863ee9bd71c575740  
  
ho!a.c 2  
2afc7a0dee4a470029a6ab40977e6e01f7b29896  
  
ho!a.c 3  
24ec4c7810126d31e445c0b863ee9bd71c575740  
  
ho!a.c 4  
81fc6b4d67dc1c4ac401dfd37c1677a8c54a6f3a5e775370dd366f4b0b573620  
  
ho!a.c 5  
3b54f0963fb51d58e02ee48e3486fdb3dfa13a25  
  
PC@DESKTOP-5II5100 MINGW64 ~  
$ shasum -a 256 ho!a.c 81fc6b4d67dc1c4ac401dfd37c1677a8c54a6f3a5e775370dd366f4b0b573620 ho!a.c cl  
ear  
81fc6b4d67dc1c4ac401dfd37c1677a8c54a6f3a5e775370dd366f4b0b573620 *ho!a.c  
shasum: 81fc6b4d67dc1c4ac401dfd37c1677a8c54a6f3a5e775370dd366f4b0b573620:  
81fc6b4d67dc1c4ac401dfd37c1677a8c54a6f3a5e775370dd366f4b0b573620 *ho!a.c
```