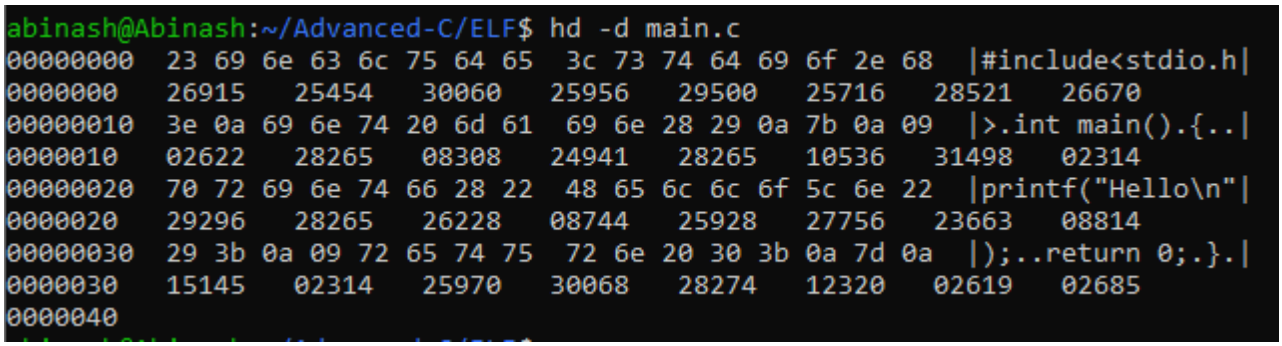
Steps to find the meaning of the binary data present in the ELF file :

```
/* hello.c */
#include<stdio.h>
int main ()
{
    printf("Hello\n");
    return 0;
}
```

Binary format of the above program:



```
abinash@Abinash:~/Advanced-C/ELF$ hd -d main.c
00000000  23 69 6e 63 6c 75 64 65  3c 73 74 64 69 6f 2e 68  |#include<stdio.h|
0000000   26915   25454   30060   25956   29500   25716   28521   26670
00000010  3e 0a 69 6e 74 20 6d 61  69 6e 28 29 0a 7b 0a 09  |>.int main().{..|
0000010   02622   28265   08308   24941   28265   10536   31498   02314
00000020  70 72 69 6e 74 66 28 22  48 65 6c 6c 6f 5c 6e 22  |printf("Hello\n"|
0000020   29296   28265   26228   08744   25928   27756   23663   08814
00000030  29 3b 0a 09 72 65 74 75  72 6e 20 30 3b 0a 7d 0a  |);..return 0;.}.|
0000030   15145   02314   25970   30068   28274   12320   02619   02685
0000040
```

Note:

In the above binary format, in the first line it has some hexadecimal numbers like 23 69 6e 63 6c 6c 75 64 65 3c 73 74 64 69 6f 2e 68

In this case, 23 refers to #, 69 refers to i, 6e refers to n etc. Likewise, every word from the code is converted from hexadecimal value.

00000000 – 00000030 represents the line number.

In this way we can find the code from the binary data and vice versa.

When we use the command (hexdump –Cv main > main.dump)

```
00000000  7f 45 4c 46 02 01 01 00  00 00 00 00 00 00 00 00  |.ELF............|
00000010  03 00 3e 00 01 00 00 00  60 10 00 00 00 00 00 00  |..>.....`.......|

00002000  01 00 02 00 48 65 6c 6c  6f 00 00 00 01 1b 03 3b  |....Hello......;|
```

01 – SOH (Start of heading)

02 – STX (Start of text)

48 – H

65 – e

6c – l **These are the hexadecimal values for the output “hello”.**

6c – l

6f – o

3b - ;

In first line, 7f is the magic number and it is a fixed byte. 45, 4c, 46 refers to e l f respectively.

The section data starts approximately from line 00004160.

The main.c filename is present in 00003f10 line no in the hex code.

ASCII Commands:

```
Dec Hex    Dec Hex    Dec Hex  Dec Hex  Dec Hex  Dec Hex   Dec Hex   Dec Hex
  0 00 NUL  16 10 DLE  32 20    48 30 0  64 40 @  80 50 P   96 60 `  112 70 p
  1 01 SOH  17 11 DC1  33 21 !  49 31 1  65 41 A  81 51 Q   97 61 a  113 71 q
  2 02 STX  18 12 DC2  34 22 "  50 32 2  66 42 B  82 52 R   98 62 b  114 72 r
  3 03 ETX  19 13 DC3  35 23 #  51 33 3  67 43 C  83 53 S   99 63 c  115 73 s
  4 04 EOT  20 14 DC4  36 24 $  52 34 4  68 44 D  84 54 T  100 64 d  116 74 t
  5 05 ENQ  21 15 NAK  37 25 %  53 35 5  69 45 E  85 55 U  101 65 e  117 75 u
  6 06 ACK  22 16 SYN  38 26 &  54 36 6  70 46 F  86 56 V  102 66 f  118 76 v
  7 07 BEL  23 17 ETB  39 27 '  55 37 7  71 47 G  87 57 W  103 67 g  119 77 w
  8 08 BS   24 18 CAN  40 28 (  56 38 8  72 48 H  88 58 X  104 68 h  120 78 x
  9 09 HT   25 19 EM   41 29 )  57 39 9  73 49 I  89 59 Y  105 69 i  121 79 y
 10 0A LF   26 1A SUB  42 2A *  58 3A :  74 4A J  90 5A Z  106 6A j  122 7A z
 11 0B VT   27 1B ESC  43 2B +  59 3B ;  75 4B K  91 5B [  107 6B k  123 7B {
 12 0C FF   28 1C FS   44 2C ,  60 3C <  76 4C L  92 5C \  108 6C l  124 7C |
 13 0D CR   29 1D GS   45 2D -  61 3D =  77 4D M  93 5D ]  109 6D m  125 7D }
 14 0E SO   30 1E RS   46 2E .  62 3E >  78 4E N  94 5E ^  110 6E n  126 7E ~
 15 0F SI   31 1F US   47 2F /  63 3F ?  79 4F O  95 5F _  111 6F o  127 7F DEL
abinash@Abinash:~/Advanced-C/ELF$
```