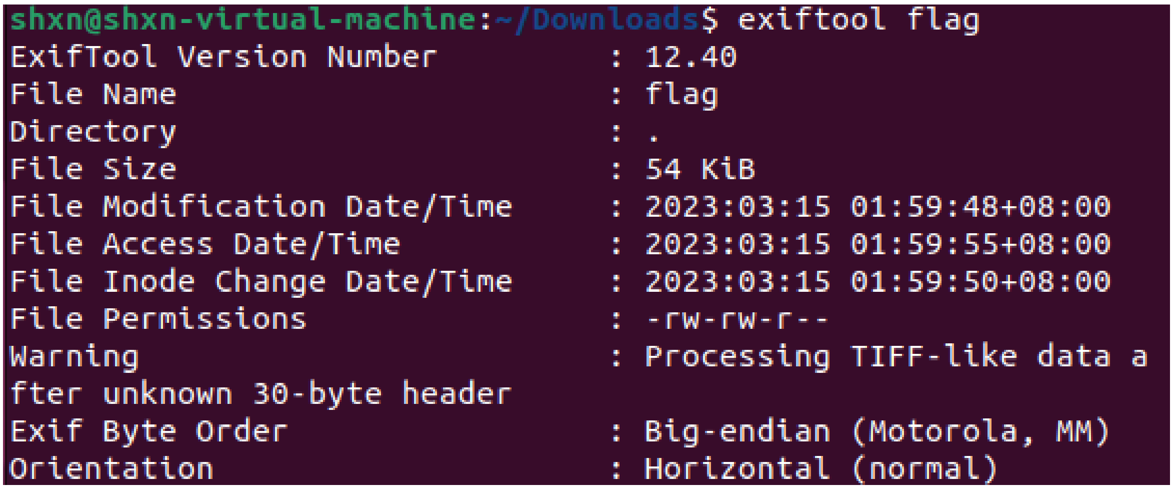
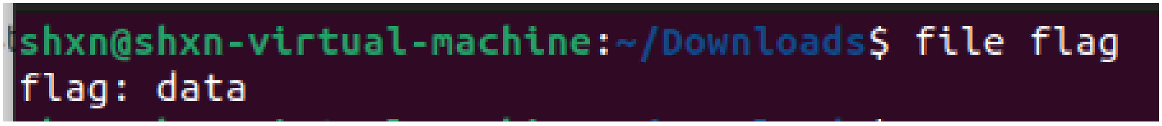
Approach to capturing the flag:

1. Inspection of file properties: We began by examining the properties of the flag file, which showed that it was of "Binary (application/octet-stream)" type. Attempts to open the file on a text editor resulted in gibberish.
2. 2. Checking file details: I suspected that the file might be an image that had been corrupted, so I ran a command to check its details. The command warned us that the file contained "TIFF-like data after unknown 30-byte header," indicating that it was indeed corrupted.

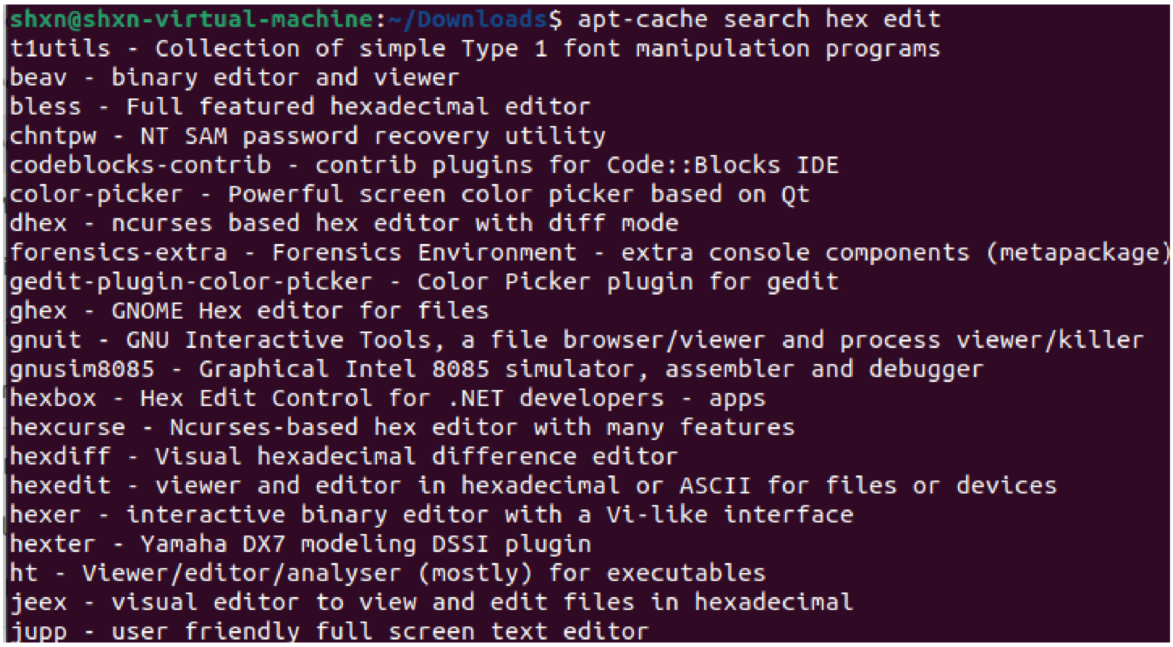


```
shxn@shxn-virtual-machine:~/Downloads$ exiftool flag
ExifTool Version Number         : 12.40
File Name                       : flag
Directory                       : .
File Size                       : 54 KiB
File Modification Date/Time     : 2023:03:15 01:59:48+08:00
File Access Date/Time           : 2023:03:15 01:59:55+08:00
File Inode Change Date/Time     : 2023:03:15 01:59:50+08:00
File Permissions                : -rw-rw-r--
Warning                         : Processing TIFF-like data a
fter unknown 30-byte header
Exif Byte Order                 : Big-endian (Motorola, MM)
Orientation                     : Horizontal (normal)
```



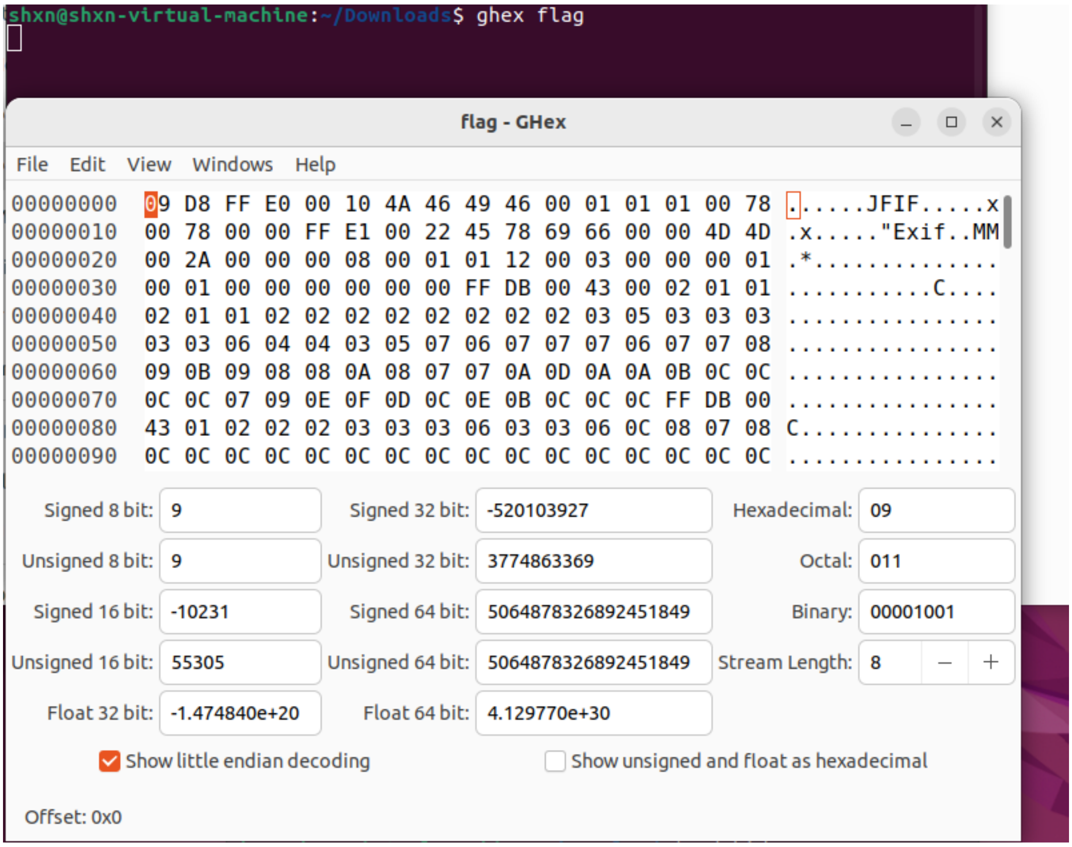
```
shxn@shxn-virtual-machine:~/Downloads$ file flag
flag: data
```

3. Inspecting file header: Since the file was not a valid image, we decided to inspect its header to identify the issue. The following command is used to identify hex editors that we could use.



```
shxn@shxn-virtual-machine:~/Downloads$ apt-cache search hex edit
t1utils - Collection of simple Type 1 font manipulation programs
beav - binary editor and viewer
bless - Full featured hexadecimal editor
chntpw - NT SAM password recovery utility
codeblocks-contrib - contrib plugins for Code::Blocks IDE
color-picker - Powerful screen color picker based on Qt
dhex - ncurses based hex editor with diff mode
forensics-extra - Forensics Environment - extra console components (metapackage)
gedit-plugin-color-picker - Color Picker plugin for gedit
ghex - GNOME Hex editor for files
gnuit - GNU Interactive Tools, a file browser/viewer and process viewer/killer
gnusim8085 - Graphical Intel 8085 simulator, assembler and debugger
hexbox - Hex Edit Control for .NET developers - apps
hexcurse - Ncurses-based hex editor with many features
hexdiff - Visual hexadecimal difference editor
hexedit - viewer and editor in hexadecimal or ASCII for files or devices
hexer - interactive binary editor with a Vi-like interface
hexter - Yamaha DX7 modeling DSSI plugin
ht - Viewer/editor/analyser (mostly) for executables
jeex - visual editor to view and edit files in hexadecimal
jupp - user friendly full screen text editor
```

4. From the above list, I installed GHex with the command $sudo apt-get install ghex. Then I ran GHex followed by the file name, which opened the application and displayed the file's hex code



```
shxn@shxn-virtual-machine:~/Downloads$ ghex flag
```

flag - GHex

```
00000000  09 D8 FF E0 00 10 4A 46 49 46 00 01 01 01 00 78  ......JFIF.....x
00000010  00 78 00 00 FF E1 00 22 45 78 69 66 00 00 4D 4D  .x....."Exif..MM
00000020  00 2A 00 00 00 08 00 01 01 12 00 03 00 00 00 01  .*..............
00000030  00 01 00 00 00 00 00 00 FF DB 00 43 00 02 01 01  ...........C....
00000040  02 01 01 02 02 02 02 02 02 02 02 03 05 03 03 03  ................
00000050  03 03 06 04 04 03 05 07 06 07 07 07 06 07 07 08  ................
00000060  09 0B 09 08 08 0A 08 07 07 0A 0D 0A 0A 0B 0C 0C  ................
00000070  0C 0C 07 09 0E 0F 0D 0C 0E 0B 0C 0C 0C FF DB 00  ................
00000080  43 01 02 02 02 03 03 03 06 03 03 06 0C 08 07 08  C...............
00000090  0C 0C 0C 0C 0C 0C 0C 0C 0C 0C 0C 0C 0C 0C 0C 0C  ................
```

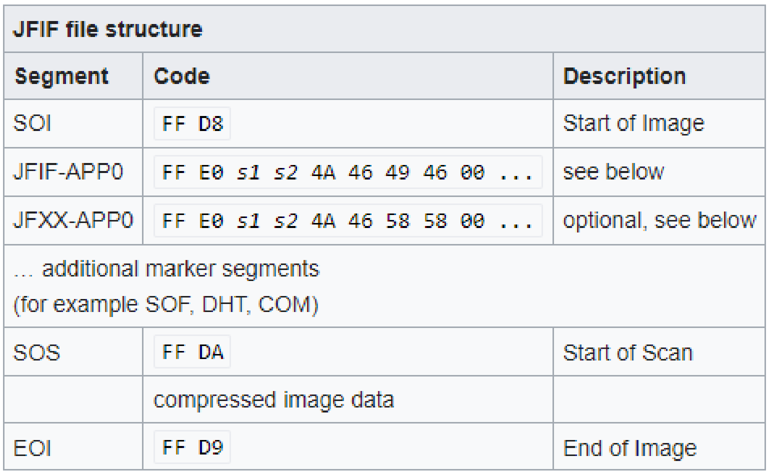
Signed 8 bit: 9
Signed 32 bit: -520103927
Hexadecimal: 09
Unsigned 8 bit: 9
Unsigned 32 bit: 3774863369
Octal: 011
Signed 16 bit: -10231
Signed 64 bit: 5064878326892451849
Binary: 00001001
Unsigned 16 bit: 55305
Unsigned 64 bit: 5064878326892451849
Stream Length: 8
Float 32 bit: -1.474840e+20
Float 64 bit: 4.129770e+30

☑ Show little endian decoding
□ Show unsigned and float as hexadecimal

Offset: 0x0

5. Correcting image header: From the hex code, we identified the first row as the image header and searched for the JPEG File Interchange Format markers.

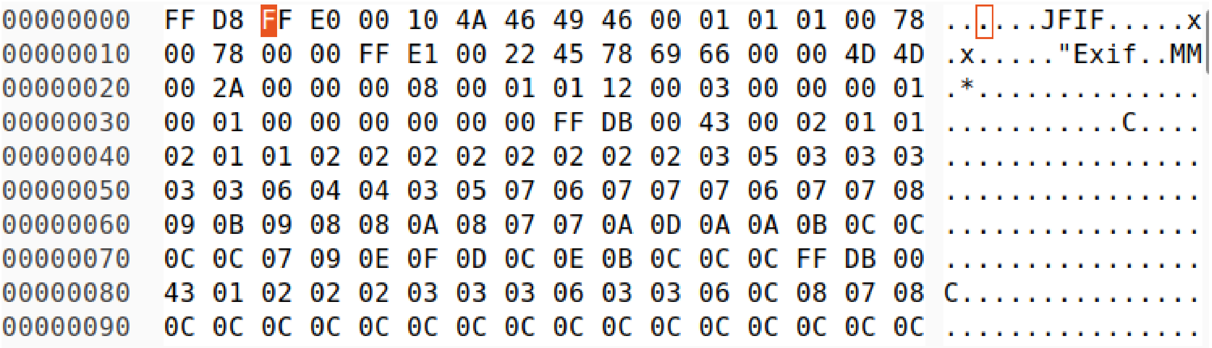
   Notice that the Start Of Image (SOI) marker was invalid and needed correction.



JFIF file structure

Segment	Code	Description
SOI	`FF D8`	Start of Image
JFIF-APP0	`FF E0 s1 s2 4A 46 49 46 00 ...`	see below
JFXX-APP0	`FF E0 s1 s2 4A 46 58 58 00 ...`	optional, see below
... additional marker segments (for example SOF, DHT, COM)		
SOS	`FF DA`	Start of Scan
	compressed image data	
EOI	`FF D9`	End of Image

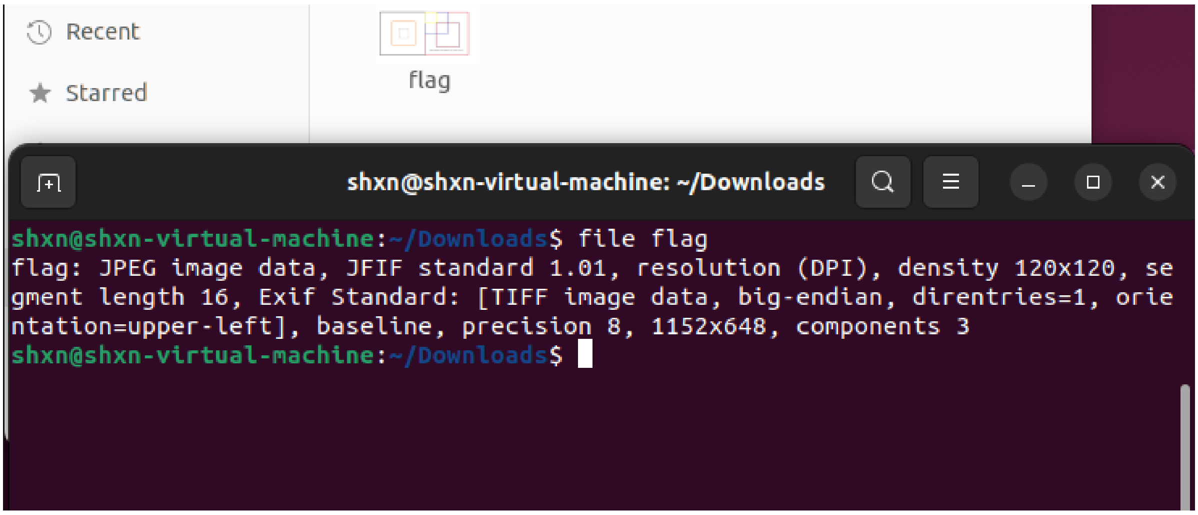
   Changed it to FF E0 as follows, and save the file:



```
00000000  FF D8 FF E0 00 10 4A 46 49 46 00 01 01 01 00 78  ......JFIF.....x
00000010  00 78 00 00 FF E1 00 22 45 78 69 66 00 00 4D 4D  .x....."Exif..MM
00000020  00 2A 00 00 00 08 00 01 01 12 00 03 00 00 00 01  .*..............
00000030  00 01 00 00 00 00 00 00 FF DB 00 43 00 02 01 01  ...........C....
00000040  02 01 01 02 02 02 02 02 02 02 02 03 05 03 03 03  ................
00000050  03 03 06 04 04 03 05 07 06 07 07 07 06 07 07 08  ................
00000060  09 0B 09 08 08 0A 08 07 07 0A 0D 0A 0A 0B 0C 0C  ................
00000070  0C 0C 07 09 0E 0F 0D 0C 0E 0B 0C 0C 0C FF DB 00  ................
00000080  43 01 02 02 02 03 03 03 06 03 03 06 0C 08 07 08  C...............
00000090  0C 0C 0C 0C 0C 0C 0C 0C 0C 0C 0C 0C 0C 0C 0C 0C  ................
```

Results:

After saving the file, it is observed that its type had changed to JPEG file, indicating that the correction had been successful. Upon opening the JPEG file, you will see the flag.



```
shxn@shxn-virtual-machine:~/Downloads$ file flag
flag: JPEG image data, JFIF standard 1.01, resolution (DPI), density 120x120, se
gment length 16, Exif Standard: [TIFF image data, big-endian, direntries=1, orie
ntation=upper-left], baseline, precision 8, 1152x648, components 3
shxn@shxn-virtual-machine:~/Downloads$ 
```

Conclusion:

In conclusion, the approach to capturing the flag involved a series of steps, including inspection of file properties, checking file details, inspecting the file header, and correcting the image header.