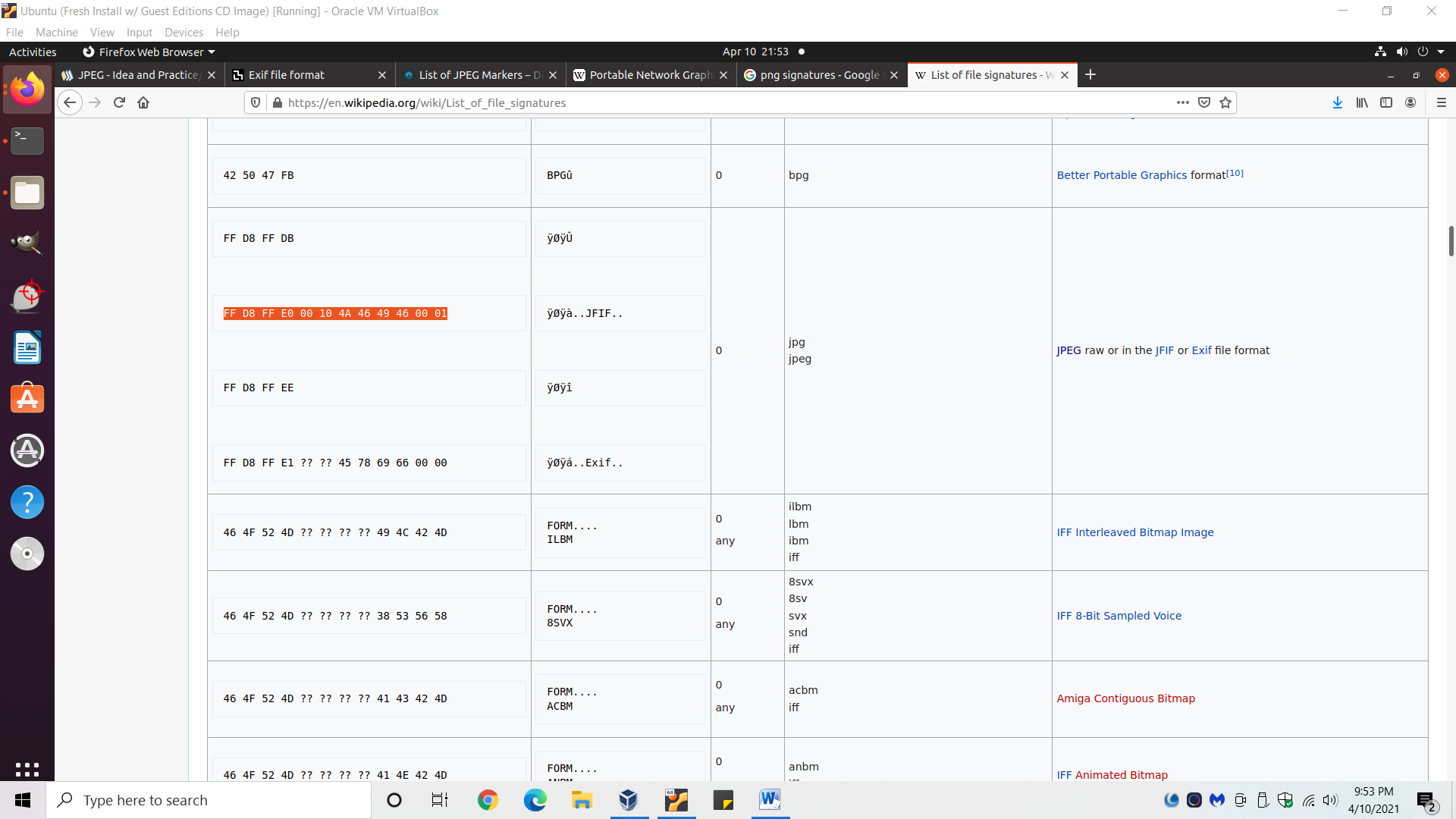
**Magic Bytes (Medium)**

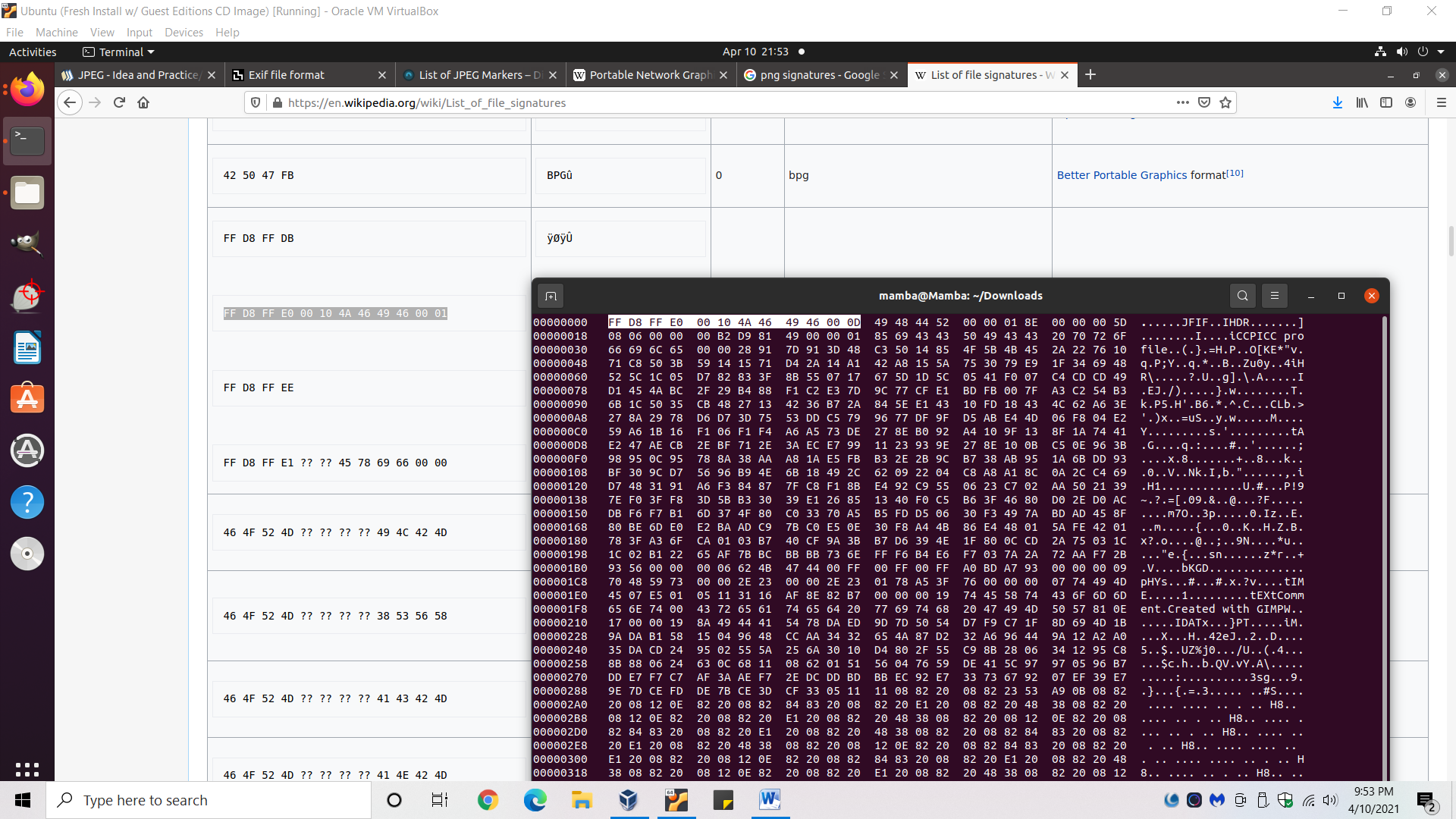
**This file appears to be changed in some way. Can you recover the original?**

**What is the original file type? (25 pts)**

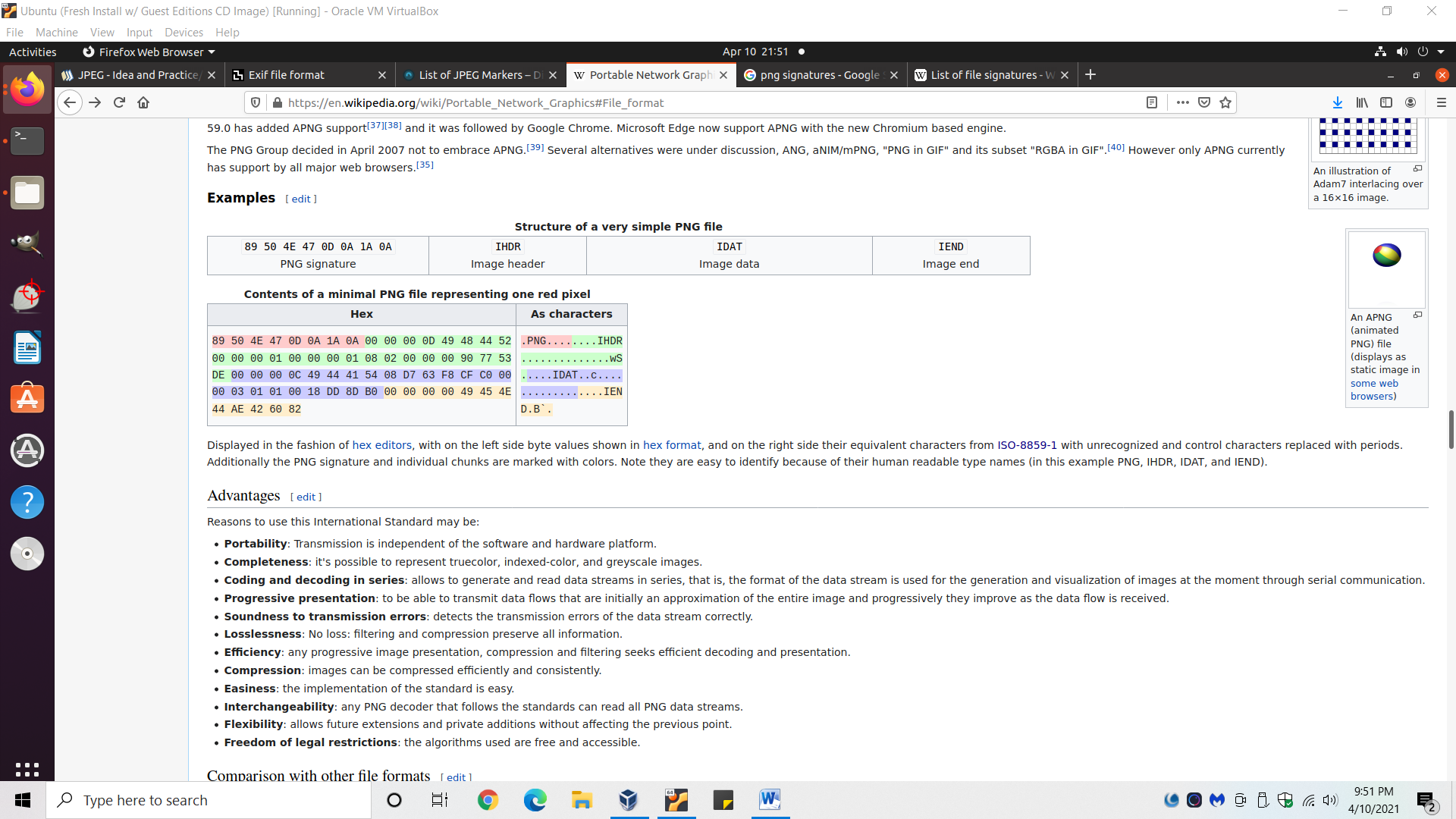
* There’s a hint is in the title of the challenge with the keywords “magic bytes.” Magic bytes are a list of file signatures, which are data used to identify or verify the content of a file. Such signatures are also known as magic numbers or Magic Bytes.
  + You can look at these magic bytes / magic numbers by viewing the hex values of the file.
* Download the *hexedit* tool in the command line by typing in *sudo apt-get install hexedit* if you don’t already have the tool installed.
  + You can use any hex editor tool of your choice, but I chose to use *hexedit*.
* Use *cd* to jump to the directory where the jpeg is.
* Type in *hexedit flag.jpeg*.
* Notice the header (top of the hex dump) has a jpeg signature (and contains evidence of a previous png signature), and scrolling all the way to the bottom the hex is formatted to be a png!

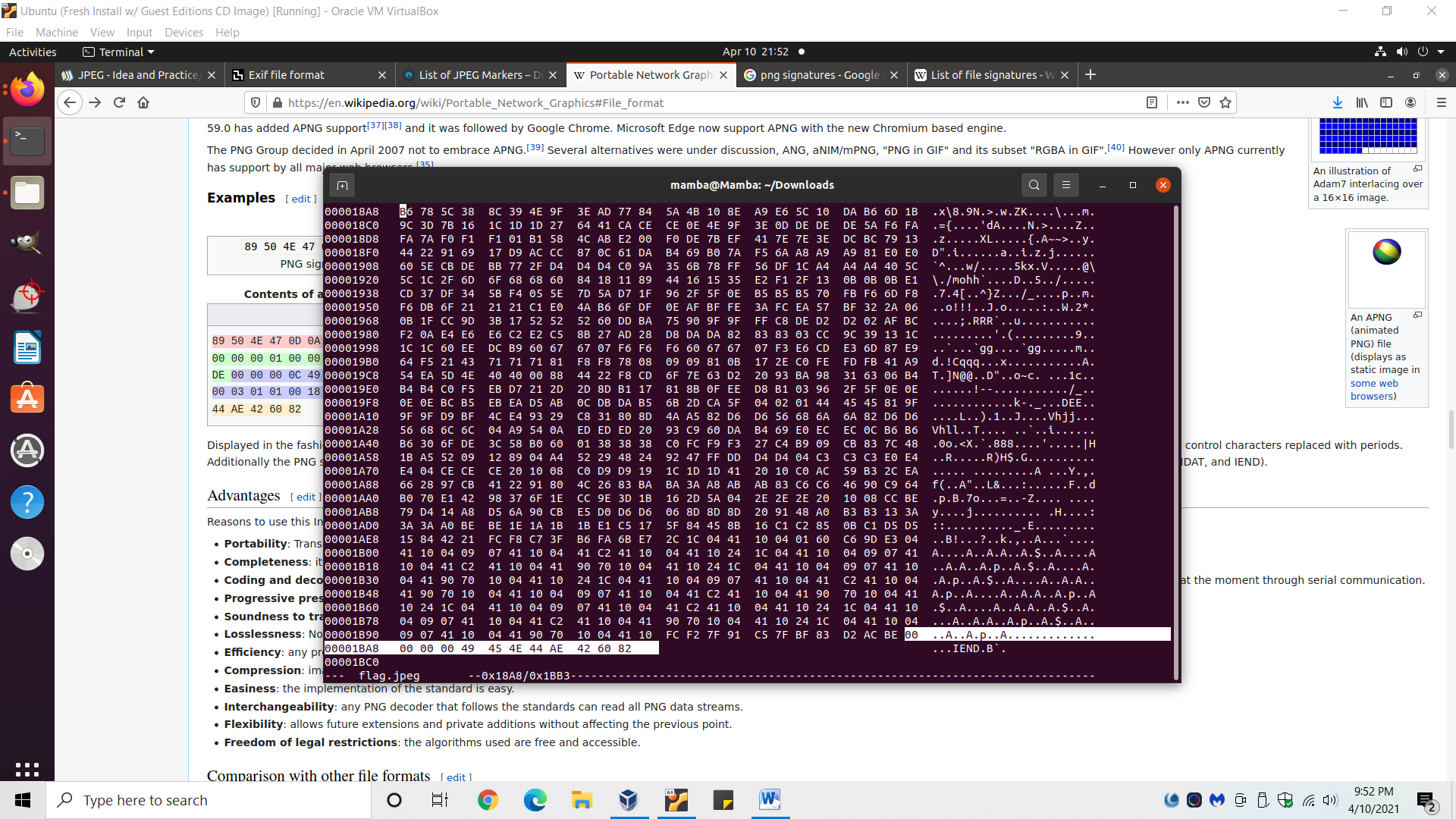


Various file signatures for a jpeg; the one used by flag.jpeg is highlighted in orange.



Screenshot from the terminal. The file header is formatted exactly to the jpeg signature above, except for the last digit (which shows evidence of a previous png signature) which controls the JFIF version.



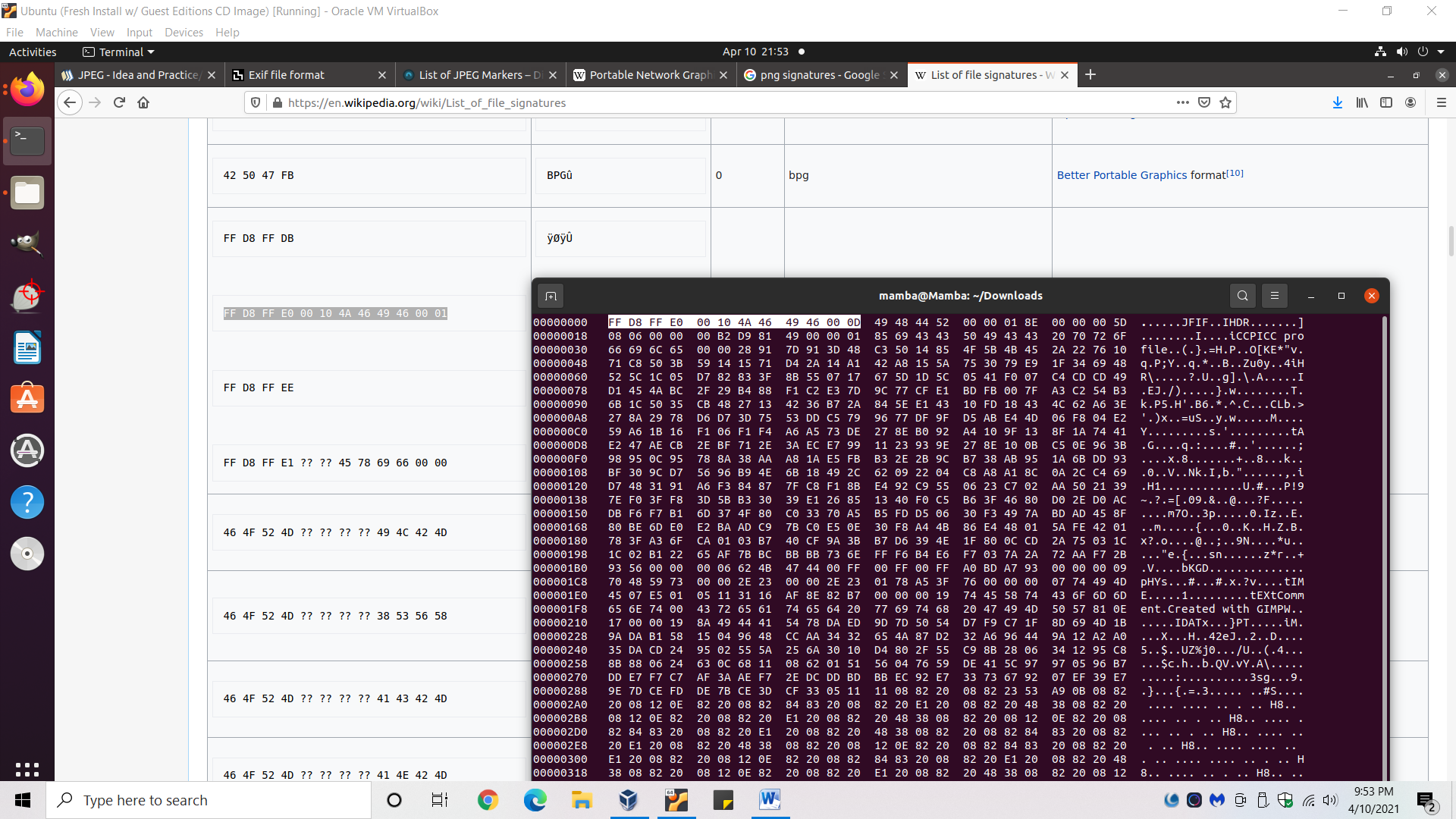


Screenshot of the terminal that shows the flag.jpeg ending hex values perfectly match the IEND (yellow highlight in the above picture) of a png.

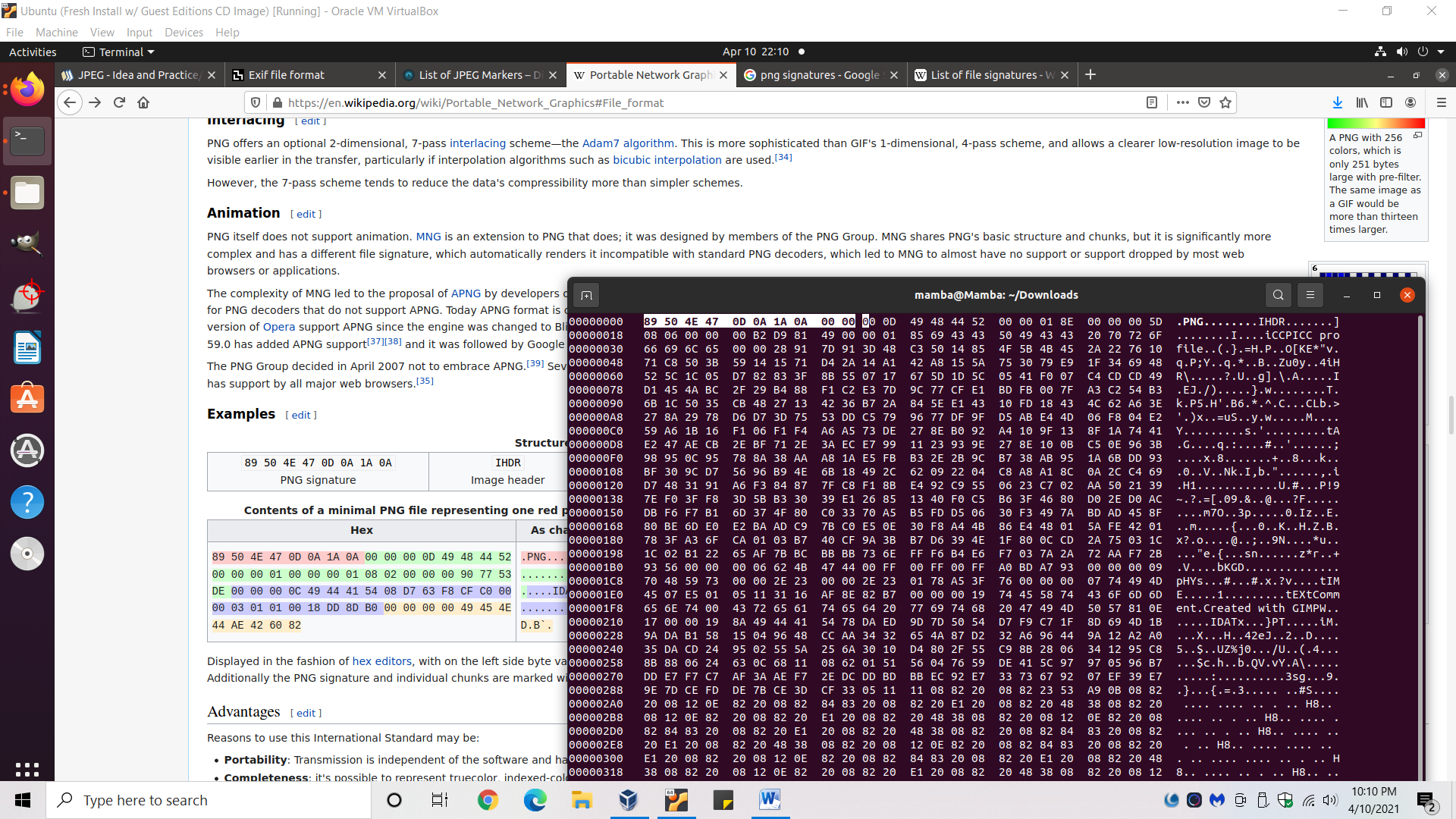
* Answer: png

**What is the flag? (75 pts)**

* In the terminal, type in *hexedit flag.jpeg* (if not already done).
* As previously seen, the majority of the hex is formatted to be a .png. The only reason it appears to be a .jpeg by programs is because the header is in the .jpeg format; it’s as if a .png is hiding behind the façade of a .jpeg.
* Format the header to be that of a PNG. The highlighted text is what has been chnaged.
  + Notice the ISO/IEC 8859-1 text (gibbersih on the right-hand side) changes from “JFIF” to “PNG.”



Original header with a .jpeg signature.



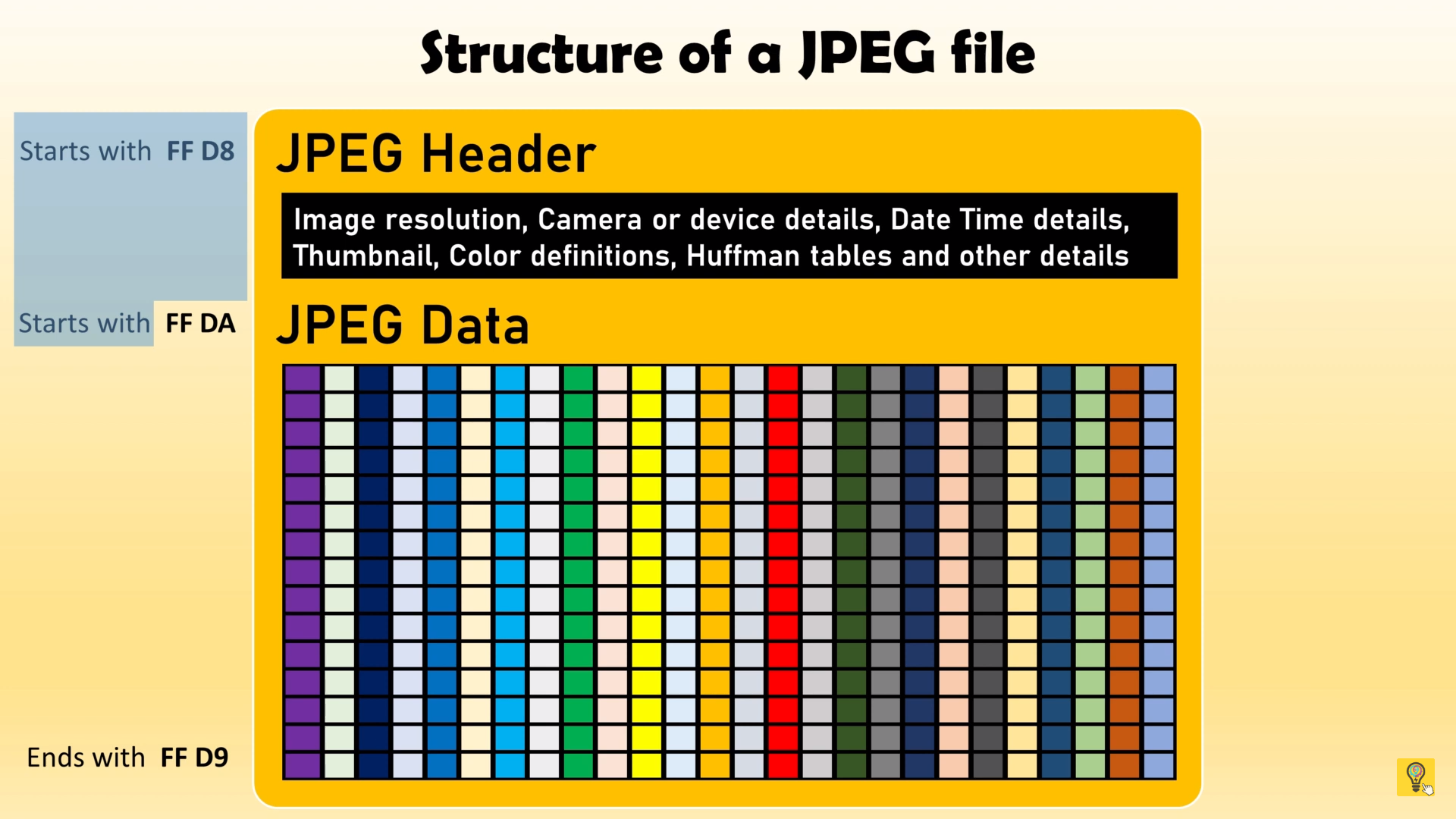
Modified header to be have the signature of a .png. These values were referenced from the the red highlighted text in the .png breakdown picture above.

* Hold [Ctrl + x] to save the changes.
* You will be prompted with “Save changes (Yes/No/Cancel) ?” and hit [y].
* Rename the file to have a .png extension. I.e. the original file name was “flag.jpeg,” but you need to rename it to “flag.png.”
  + Go to the flag.jpeg file location and right-click it.
* You can now open the file to view the flag.
* Answer: SKY-DSFG-5792

**References**

\*Some of these references deal with jpeg signatures and other material, which may be helpful to reference for future challenges dealing with similar topics!

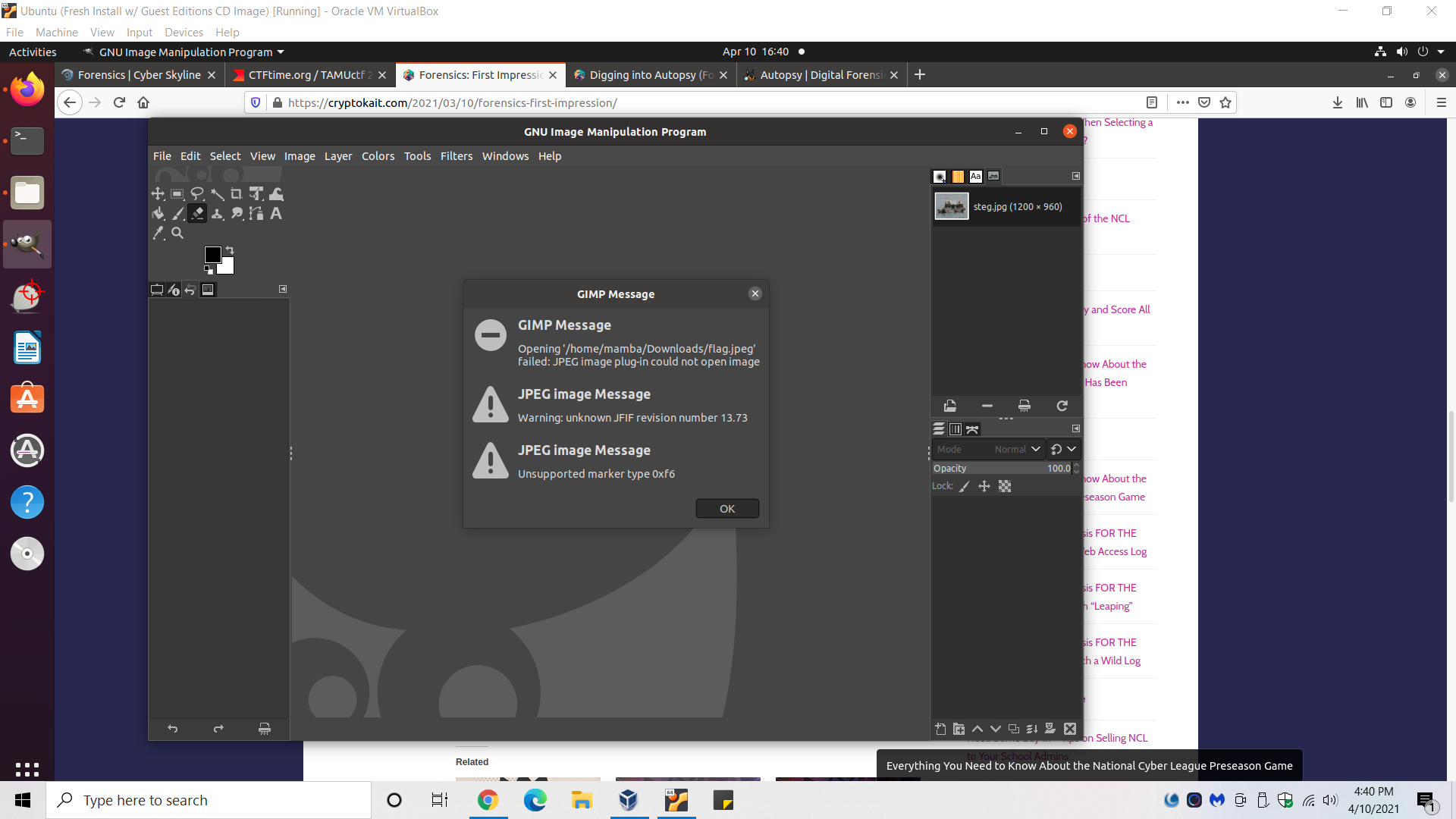
* HexEdit commands tutorial: <https://www.youtube.com/watch?v=Jhe7wiAoc-w>
* Wikipedia on PNG w/ some hex marker values: https://en.wikipedia.org/wiki/Portable\_Network\_Graphics#File\_format
* Magic bytes Wikipedia page w/ common headers for lots of file types: <https://en.wikipedia.org/wiki/List_of_file_signatures>
* Working with magix #’s in Linux: <https://www.geeksforgeeks.org/working-with-magic-numbers-in-linux/>
* What are magic bytes and how to exploit: <https://medium.com/@d.harish008/what-is-a-magic-byte-and-how-to-exploit-1e286da1c198>
* NCL forensics tips: https://cryptokait.com/2021/03/10/forensics-first-impression/
* List of JPEG markers: <https://www.disktuna.com/list-of-jpeg-markers/>
* Semi-helpful video where the below screenshot was taken from: <https://www.youtube.com/watch?v=L2DoCDROcuY>



\*Below is scratch work and observations from the research I put into this. Although it’s not needed to solve this challenge, there are observations to failed attempts that helped me learn about hex values, signatures, etc. The below material is more for me, in case I need to reference my scratchwork in the future.

**What is the flag? (75 pts)**

* When using GIMP to try and open the jpeg, the following errors are seen.
  + Note: GIMP provides more detailed error analysis than other programs.



* Hint is in the title; keyword “magic bytes”!
  + Magic Bytes: a list of file signatures, data used to identify or verify the content of a file. Such signatures are also known as magic numbers or Magic Bytes. Many file formats are not intended to be read as text. If such a file is accidentally viewed as a text file, its contents will be unintelligible.
* To look at the jpeg’s magic bytes, we ned to look at the hex values by using a hex-editor.
  + I use the Debian based package *hexeditor* downloaded from the terminal via the command *sudo apt-get install hexeditor*.
  + HexEdit commands tutorial: <https://www.youtube.com/watch?v=Jhe7wiAoc-w>
* Type in *hexedit flag.jpeg*  to open hex values.
* Answer: SKY-DSFG-5792

**Notable Potential Problems & Relevant Points**

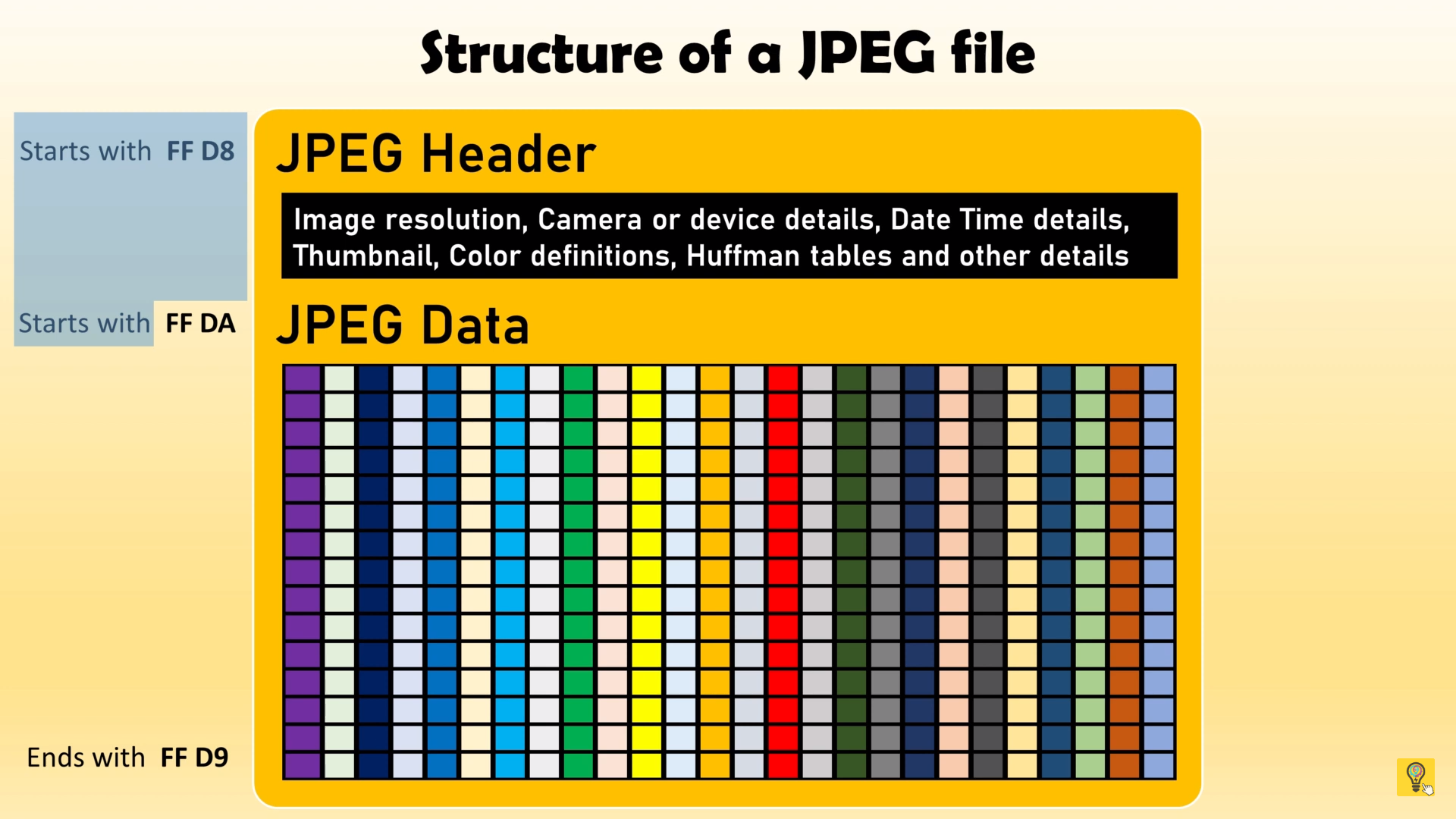
1. The header’s last digit is ‘D’ instead of ‘1’
   1. Magic Byte header reference table: https://en.wikipedia.org/wiki/List\_of\_file\_signatures
      1. FF D8 FF E0 00 10 4A 46 49 46 00 0D = original jpeg header
      2. FF D8 FF E0 00 10 4A 46 49 46 00 01 = standard jpeg header
      3. \*changing this digit changes the JFIF version# from 13.73 to 1.73
2. There’s no FFD9 marker to indicate the end of the JPEG!
   1. Upon looking at other JPEG’s, FFD9 is always in the place of the last hex slot; at the very end of the hex dump.
   2. The ending is formatted for a PNG!
3. No SOF or SOS markers.
   1. Reference: : <https://www.disktuna.com/list-of-jpeg-markers/>

**Possible Approaches to the Solution**

1. Use *hexedit* to properly align the hex values according to jpeg standards.
2. The original file was a .PNG, so maybe the hex values are already more setup to a PNG format (i.e. only the header was converted to jpeg, so that’s why it’s perceived to be a jpeg). So go through the hex values with *hexedit* to align them to PNG standards.
3. Try and pull the thumbnail of the photo out of the header or hex values somewhere; i.e. you don’t have to fix the whole file, just enough to grab the thumbnail.

**References**

* HexEdit commands tutorial: <https://www.youtube.com/watch?v=Jhe7wiAoc-w>
* List of JPEG markers: <https://www.disktuna.com/list-of-jpeg-markers/>
* Magic bytes Wikipedia page w/ common headers for lots of file types: <https://en.wikipedia.org/wiki/List_of_file_signatures>
* Semi-helpful video where the below screenshot was taken from: <https://www.youtube.com/watch?v=L2DoCDROcuY>



* Working with magix #’s in Linux: <https://www.geeksforgeeks.org/working-with-magic-numbers-in-linux/>
* What are magic bytes and how to exploit: <https://medium.com/@d.harish008/what-is-a-magic-byte-and-how-to-exploit-1e286da1c198>
* Wikipedia on PNG w/ some hex marker values: https://en.wikipedia.org/wiki/Portable\_Network\_Graphics#File\_format
* NCL forensics tips: https://cryptokait.com/2021/03/10/forensics-first-impression/

**Scratch Work**

JPEG Marker List: <https://www.disktuna.com/list-of-jpeg-markers/>

0xf6 orog = 38aa

0x219 original hex# =

FF DA = 261/262 AEB/AEC 3778/3779

FF D9 = last

FF7D?? 0x98e

No FFD9 included in original.

No SOF or SOS markers in original.

Had to change last hex digit of header from D to 1…changed JFIF version.