LINUX-BASED FILE ANALYSIS TOOL

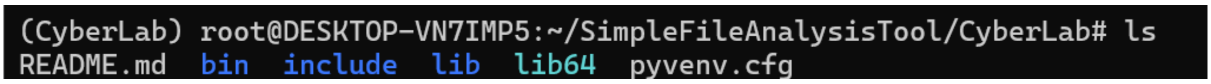
Chirag Sharma

IIT(ISM) Dhanbad

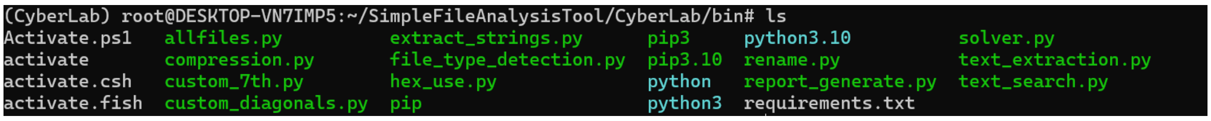
About my project

I have created a total of 11 commands from total basics which include File Type Detection, Text Extraction, Header Analysis, Steganography, File Compression and Decompression etc.

Also I have made a command of report generation which combine file type detection, extraction of readable text, Header hexadecimal representation and analysis based on it like file space, resolution, compression method etc.



```
(CyberLab) root@DESKTOP-VN7IMP5:~/SimpleFileAnalysisTool/CyberLab# ls
README.md  bin  include  lib  lib64  pyvenv.cfg
```

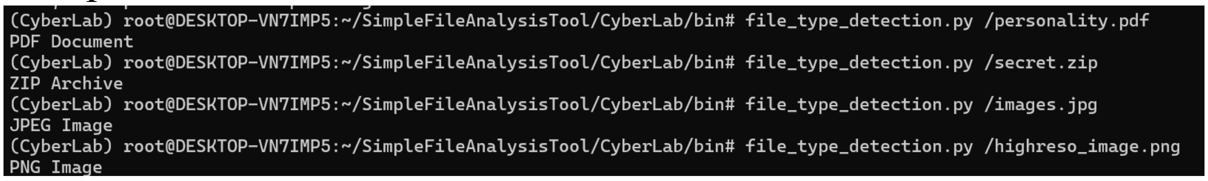


```
(CyberLab) root@DESKTOP-VN7IMP5:~/SimpleFileAnalysisTool/CyberLab/bin# ls
Activate.ps1   allfiles.py          extract_strings.py      pip3     python3.10          solver.py
activate       compression.py       file_type_detection.py  pip3.10  rename.py           text_extraction.py
activate.csh   custom_7th.py        hex_use.py              python   report_generate.py  text_search.py
activate.fish  custom_diagonals.py  pip                     python3  requirements.txt
```

Detailed explanation:

1. File Type Detection

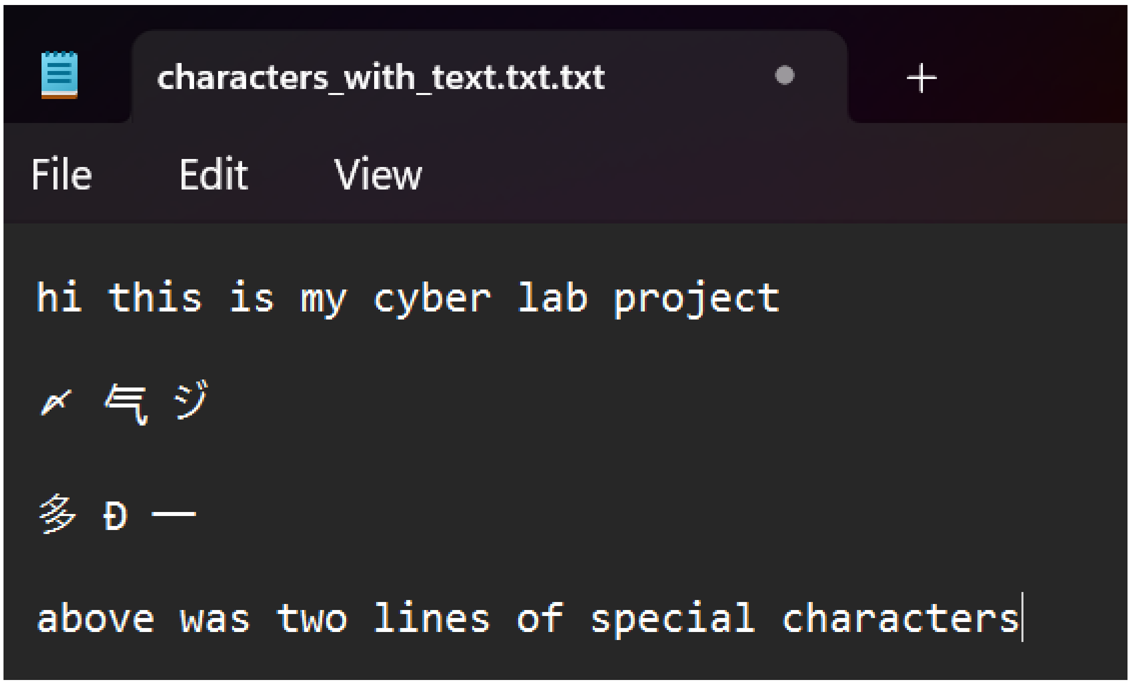
This Python script identifies file types based on their signatures and content. It checks the first 8 bytes of a file against known file signatures for formats like PNG, JPEG, ZIP, PDF, and GZIP. If no match is found, it attempts to classify the file as a text file, binary file, hex dump, or unknown. The script uses command-line arguments to specify the file path and provides error handling for missing files or unexpected issues.



```
(CyberLab) root@DESKTOP-VN7IMP5:~/SimpleFileAnalysisTool/CyberLab/bin# file_type_detection.py /personality.pdf
PDF Document
(CyberLab) root@DESKTOP-VN7IMP5:~/SimpleFileAnalysisTool/CyberLab/bin# file_type_detection.py /secret.zip
ZIP Archive
(CyberLab) root@DESKTOP-VN7IMP5:~/SimpleFileAnalysisTool/CyberLab/bin# file_type_detection.py /images.jpg
JPEG Image
(CyberLab) root@DESKTOP-VN7IMP5:~/SimpleFileAnalysisTool/CyberLab/bin# file_type_detection.py /highreso_image.png
PNG Image
```

2. Text Extraction

This script reads a binary file byte by byte, retaining printable ASCII characters (32–126) and newline/carriage return characters, while ignoring others. It returns the extracted text as a string. The script also uses a command-line argument.

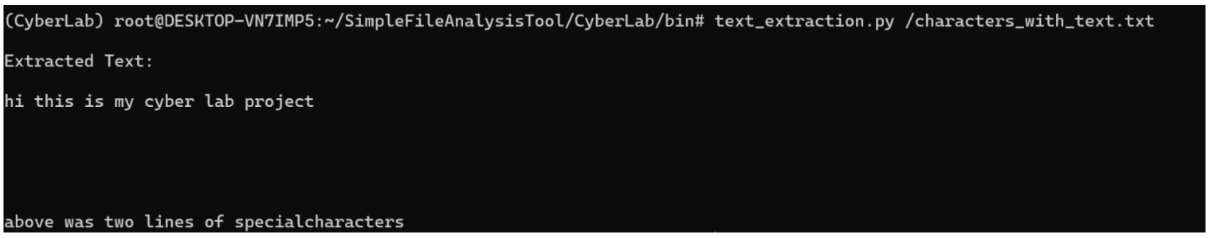


```
hi this is my cyber lab project

〆 气 ジ

多 Đ 一

above was two lines of special characters
```



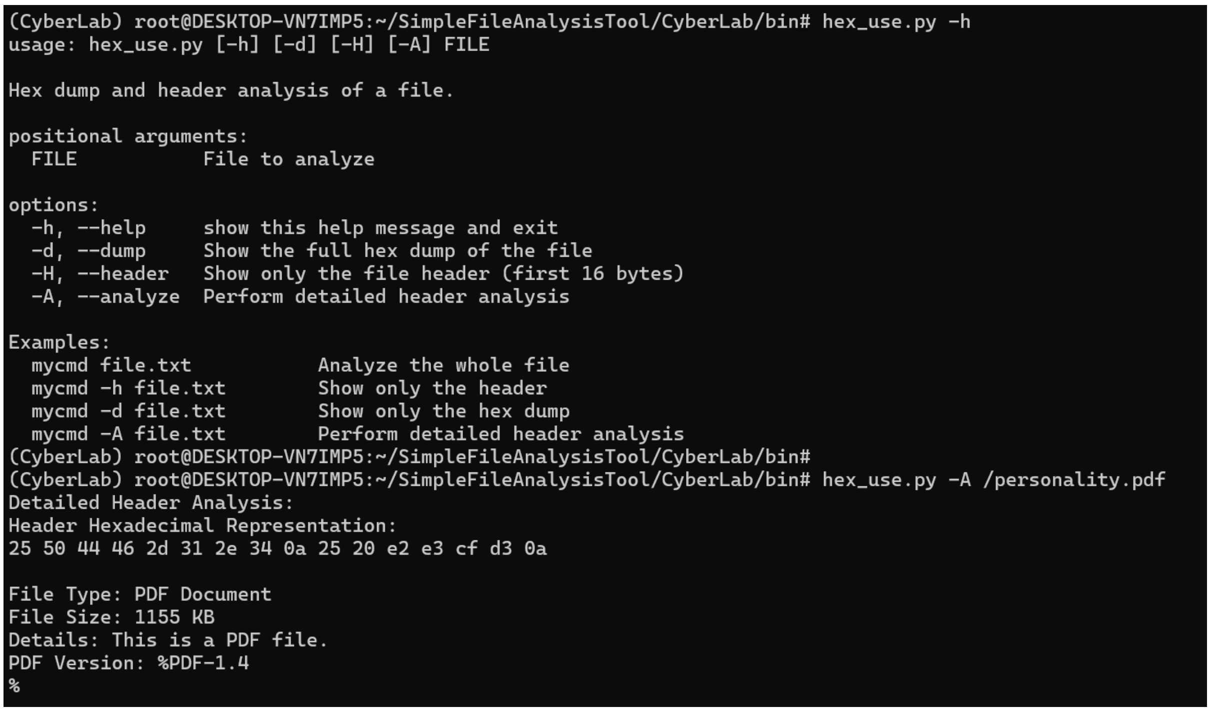
```
(CyberLab) root@DESKTOP-VN7IMP5:~/SimpleFileAnalysisTool/CyberLab/bin# text_extraction.py /characters_with_text.txt
Extracted Text:
hi this is my cyber lab project



above was two lines of specialcharacters
```

3. Hex Dump and Header Analysis

This Python script offers a comprehensive analysis of a file's contents. It generates a hex dump, reads the file's header, and determines its type by examining the first 16 bytes. The script can identify common file formats such as PNG, JPEG, PDF, ZIP, and GZIP, as well as detect plain text or unknown files. Furthermore, it provides insights into possible file structures and custom binary formats. Users have the option to specify various command-line parameters to display just the hex dump, the header, or a detailed analysis.



```
(CyberLab) root@DESKTOP-VN7IMP5:~/SimpleFileAnalysisTool/CyberLab/bin# hex_use.py -h
usage: hex_use.py [-h] [-d] [-H] [-A] FILE

Hex dump and header analysis of a file.

positional arguments:
  FILE           File to analyze

options:
  -h, --help     show this help message and exit
  -d, --dump     Show the full hex dump of the file
  -H, --header   Show only the file header (first 16 bytes)
  -A, --analyze  Perform detailed header analysis

Examples:
  mycmd file.txt           Analyze the whole file
  mycmd -h file.txt        Show only the header
  mycmd -d file.txt        Show only the hex dump
  mycmd -A file.txt        Perform detailed header analysis
(CyberLab) root@DESKTOP-VN7IMP5:~/SimpleFileAnalysisTool/CyberLab/bin#
(CyberLab) root@DESKTOP-VN7IMP5:~/SimpleFileAnalysisTool/CyberLab/bin# hex_use.py -A /personality.pdf
Detailed Header Analysis:
Header Hexadecimal Representation:
25 50 44 46 2d 31 2e 34 0a 25 20 e2 e3 cf d3 0a

File Type: PDF Document
File Size: 1155 KB
Details: This is a PDF file.
PDF Version: %PDF-1.4
%
```

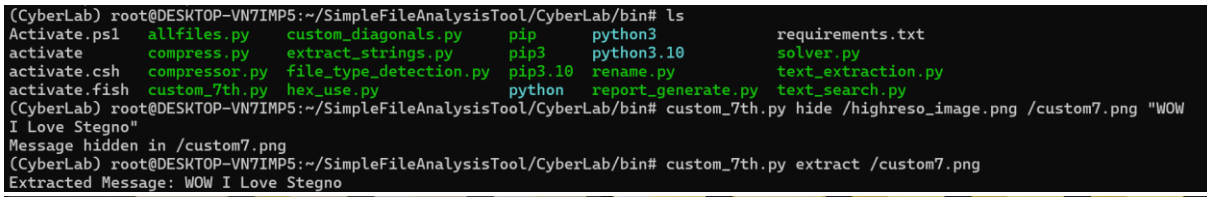
4. Steganography

I have made 3 commands for steganography solver.py, custom_diagonal.py, custom_7th.py where

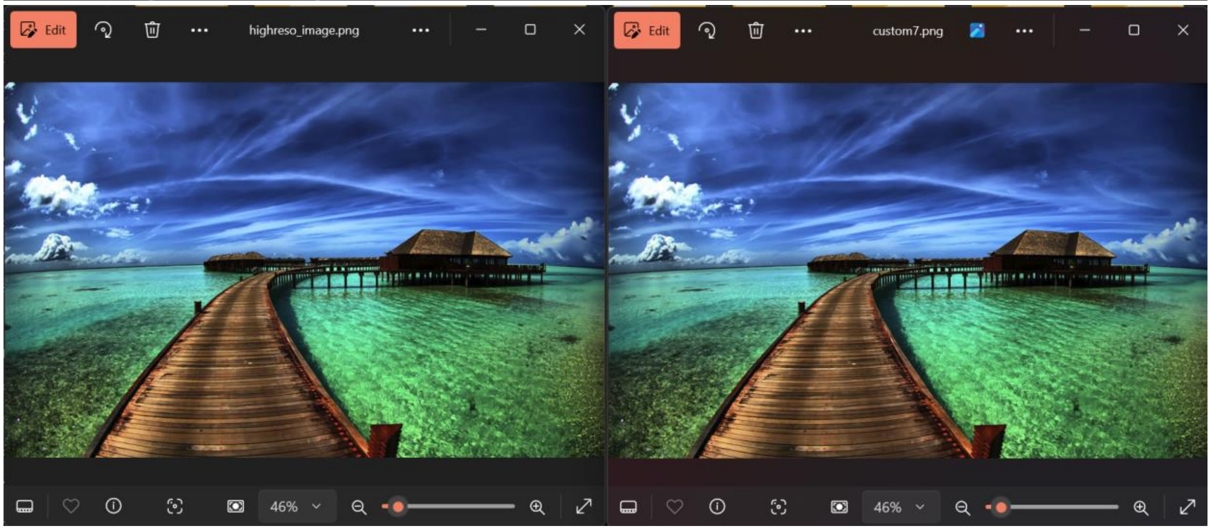
a. Solver.py: This Python script implements a simple steganography technique to hide and extract text messages in the least significant bit (LSB) of an image file. It can either hide a message inside an image (using the hide command) or extract a hidden message (using the extract command). The script reads the image as binary data, modifies the LSB of each byte to encode the message, and writes the modified image back. For extraction, it reads the LSBs to retrieve the hidden message, stopping when it detects a null terminator.

b. Custom_diagonals.py: This Python script allows hiding and extracting messages in the diagonal pixels of an image. Using the Pillow library, the script manipulates the least significant bit (LSB) of the red channel of pixels located along the diagonal of the image to encode or decode a message. The hide action embeds a binary message (terminated with a null byte) into the image, while the extract action retrieves the hidden message by reading the LSB of the diagonal pixels. The script includes functions for reading the image, modifying the pixels, and saving the modified image.

c. Custom_7th.py: This Python script conceals a message within the least significant bit (LSB) of the red channel of every 7th pixel in the image. It alters the LSB of the red channel for each pixel, beginning with the first pixel and proceeding through the image in a sequence where every 7th pixel is designated to store one bit of the message. The message is transformed into binary, and each bit is embedded into the corresponding pixel's red channel. The process concludes once the entire message, including the null terminator (00000000), has been successfully hidden.

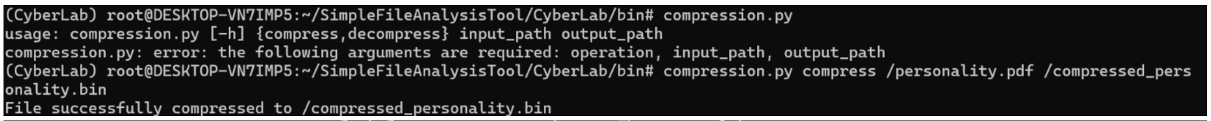


```
(CyberLab) root@DESKTOP-VN7IMP5:~/SimpleFileAnalysisTool/CyberLab/bin# ls
Activate.ps1   allfiles.py    custom_diagonals.py      pip       python3              requirements.txt
activate       compress.py    extract_strings.py       pip3      python3.10           solver.py
activate.csh   compressor.py  file_type_detection.py   pip3.10   rename.py            text_extraction.py
activate.fish  custom_7th.py  hex_use.py               python    report_generate.py   text_search.py
(CyberLab) root@DESKTOP-VN7IMP5:~/SimpleFileAnalysisTool/CyberLab/bin# custom_7th.py hide /highreso_image.png /custom7.png "WOW
I Love Stegno"
Message hidden in /custom7.png
(CyberLab) root@DESKTOP-VN7IMP5:~/SimpleFileAnalysisTool/CyberLab/bin# custom_7th.py extract /custom7.png
Extracted Message: WOW I Love Stegno
```

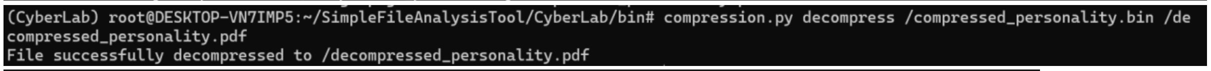


5. File Compression and Decompression

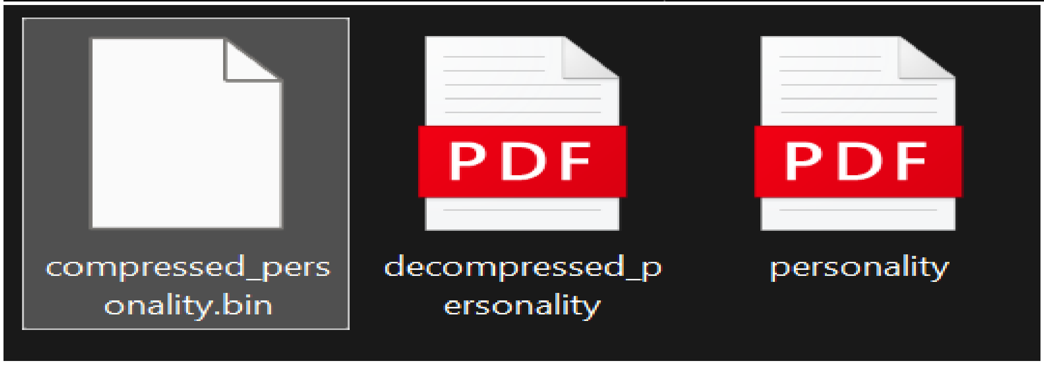
This Python script compresses and decompresses files using run-length encoding (RLE). It detects whether the file is binary or text-based, adjusts processing accordingly, and compresses consecutive identical bytes into a byte-value and count pair. For decompression, it restores the original content by repeating bytes based on the count.



```
(CyberLab) root@DESKTOP-VN7IMP5:~/SimpleFileAnalysisTool/CyberLab/bin# compression.py
usage: compression.py [-h] {compress,decompress} input_path output_path
compression.py: error: the following arguments are required: operation, input_path, output_path
(CyberLab) root@DESKTOP-VN7IMP5:~/SimpleFileAnalysisTool/CyberLab/bin# compression.py compress /personality.pdf /compressed_pers
onality.bin
File successfully compressed to /compressed_personality.bin
```

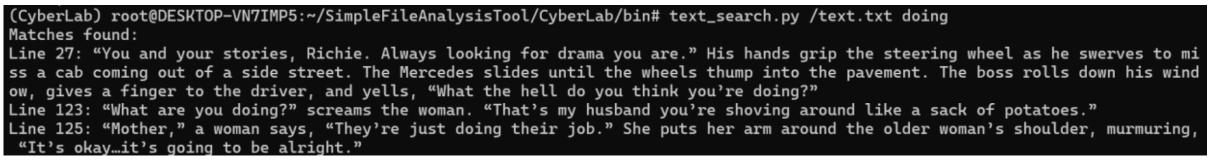


```
(CyberLab) root@DESKTOP-VN7IMP5:~/SimpleFileAnalysisTool/CyberLab/bin# compression.py decompress /compressed_personality.bin /de
compressed_personality.pdf
File successfully decompressed to /decompressed_personality.pdf
```



6. Text search

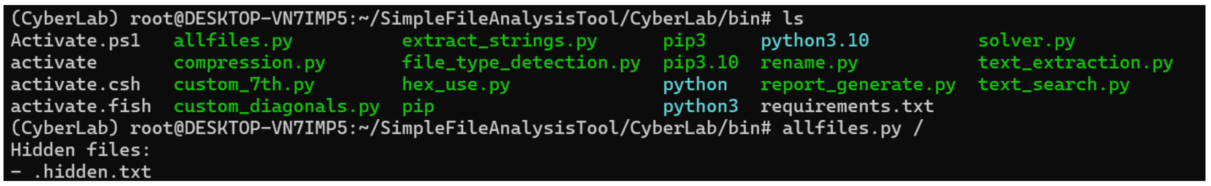
This Python script searches for a given pattern in a file, performing a case-insensitive search. It reads the file line by line, checking if the pattern exists in each line, and returns the line numbers and content where the pattern is found. It handles errors such as file not found or other exceptions.



```
(CyberLab) root@DESKTOP-VN7IMP5:~/SimpleFileAnalysisTool/CyberLab/bin# text_search.py /text.txt doing
Matches found:
Line 27: “You and your stories, Richie. Always looking for drama you are.” His hands grip the steering wheel as he swerves to mi
ss a cab coming out of a side street. The Mercedes slides until the wheels thump into the pavement. The boss rolls down his wind
ow, gives a finger to the driver, and yells, “What the hell do you think you’re doing?”
Line 123: “What are you doing?” screams the woman. “That’s my husband you’re shoving around like a sack of potatoes.”
Line 125: “Mother,” a woman says, “They’re just doing their job.” She puts her arm around the older woman’s shoulder, murmuring,
 “It’s okay…it’s going to be alright.”
```

7. Listing all hidden files

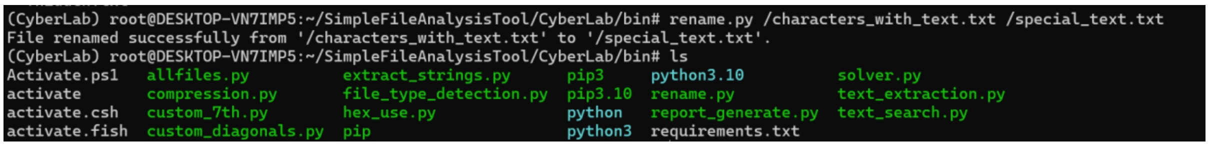
This Python script lists all hidden files (files starting with a dot) in a specified directory, defaulting to the current directory. It handles potential errors such as directory not found or lack of permission to access the directory. The script also provides an interactive command-line interface to display the hidden files or appropriate error messages.



```
(CyberLab) root@DESKTOP-VN7IMP5:~/SimpleFileAnalysisTool/CyberLab/bin# ls
Activate.ps1   allfiles.py          extract_strings.py      pip3      python3.10          solver.py
activate       compression.py       file_type_detection.py  pip3.10   rename.py           text_extraction.py
activate.csh   custom_7th.py        hex_use.py              python    report_generate.py  text_search.py
activate.fish  custom_diagonals.py  pip                     python3   requirements.txt
(CyberLab) root@DESKTOP-VN7IMP5:~/SimpleFileAnalysisTool/CyberLab/bin# allfiles.py /
Hidden files:
- .hidden.txt
```

8. Rename Files

This Python script renames a file from its current name to a new specified name. It checks for common errors such as the file not existing or lack of permissions, and provides appropriate error messages. The script uses command-line arguments to accept the current and new file names from the user.

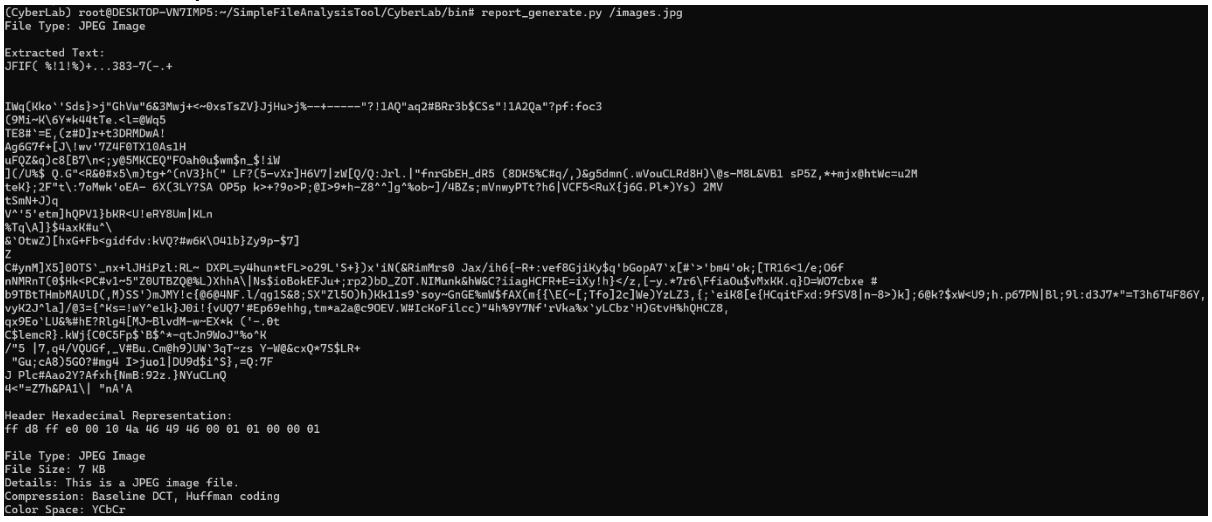


```
(CyberLab) root@DESKTOP-VN7IMP5:~/SimpleFileAnalysisTool/CyberLab/bin# rename.py /characters_with_text.txt /special_text.txt
File renamed successfully from '/characters_with_text.txt' to '/special_text.txt'.
(CyberLab) root@DESKTOP-VN7IMP5:~/SimpleFileAnalysisTool/CyberLab/bin# ls
Activate.ps1   allfiles.py          extract_strings.py       pip3     python3.10          solver.py
activate       compression.py       file_type_detection.py   pip3.10  rename.py           text_extraction.py
activate.csh   custom_7th.py        hex_use.py               python   report_generate.py  text_search.py
activate.fish  custom_diagonals.py  pip                      python3  requirements.txt
```

9. Report generation

This Python script provides a detailed analysis of a file, performing the following tasks:

1. File Type Identification: It identifies the file type based on its header signature (e.g., PNG, JPEG, ZIP, PDF, etc.).
2. Text Extraction: It extracts readable text from the file by filtering printable ASCII characters.
3. Header Analysis: It reads the first 16 bytes of the file, displays its hexadecimal representation, and provides insights such as file resolution, compression details, or specific characteristics (e.g., EXE, GZIP).
4. The script also generates a comprehensive report, including file size and any relevant metadata.



```
(CyberLab) root@DESKTOP-VN7IMP5:~/SimpleFileAnalysisTool/CyberLab/bin# report_generate.py /images.jpg
File Type: JPEG Image

Extracted Text:
JFIF( %!1!%)+...383-7(-.+


IWq(Kko`'Sds}>j"GhVw"6&3Mwj+<~0xsTsZV}JjHu>j%--+-----"?!1AQ"aq2#BRr3b$CSs"!1A2Qa"?pf:foc3
(9Mi~K\6Y*k44tTe.<l=@Wq5
TE8#`=E,(z#D]r+t3DRMDwA!
Ag6G7f+[J\!wv'7Z4F0TX10As1H
uFQZ&q)c8[B7\n<;y@5MKCEQ"FOah0u$wm$n_$!iW
](/U%$ Q.G"<R&0#x5\m)tg+^(nV3}h(" LF?(5-vXr]H6V7|zW[Q/Q:Jrl.|"fnrGbEH_dR5 (8DK5%C#q/,)&g5dmn(.wVouCLRd8H)\@s-M8L&VB1 sP5Z,*+mjx@htWc=u2M
teK};2F"t\:7oMwk'oEA- 6X(3LY?SA OP5p k>+?9o>P;@I>9*h-Z8^^]g^%ob~]/4BZs;mVnwyPTt?h6|VCF5<RuX{j6G.Pl*)Ys) 2MV
tSmN+J)q
V^'5'etm]hQPV1}bKR<U!eRY8Um|KLn
%Tq\A]}$4axK#u^\
&`OtwZ)[hxG+Fb<gidfdv:kVQ?#w6K\O41b}Zy9p-$7]
Z
C#ynM]X5]0OTS`_nx+lJHiPzl:RL~ DXPL=y4hun*tFL>o29L'S+})x'iN(&RimMrs0 Jax/ih6{-R+:vef8GjiKy$q'bGopA7`x[#`>'bm4'ok;[TR16<1/e;O6f
nNMRnT(0$Hk<PC#v1~5"Z0UTBZQ@%L)XhhA\|Ns$ioBokEFJu+;rp2)bD_ZOT.NIMunk&hW&C?iiagHCFR+E=iXy!h}</z,[-y.*7r6\FfiaOu$vMxKK.q}D=WO7cbxe #
b9TBtTHmbMAUlD(,M)SS')mJMY!c{@6@4NF.l/qg1S&8;SX"Zl5O)h)Kk11s9`soy~GnGE%mW$fAX(m{{\E(~[;Tfo]2c]We)YzLZ3,{;`eiK8[e{HCqitFxd:9fSV8|n-8>)k];6@k?$xW<U9;h.p67PN|Bl;9l:d3J7*"=T3h6T4F86Y,
vyK2J^la]/@3={^Ks=!wY^e1k}J0i!{vUQ7'#Ep69ehhg,tm*a2a@c9OEV.W#IcKoFilcc)"4h%9Y7Nf'rVka%x`yLCbz`H)GtvH%hQHCZ8,
qx9Eo`LU&%#hE?Rlg4[MJ~BlvdM-w~EX*k ('-.0t
C$lemcR}.kWj{C0C5Fp$`B$^*-qtJn9WoJ"%o^K
/"5 |7,q4/VQUGf,_V#Bu.Cm@h9)UW`3qT~zs Y-W@&cxQ*7S$LR+
 "Gu;cA8)5GO?#mg4 I>juo1|DU9d$i^S},=Q:7F
J Plc#Aao2Y?Afxh{NmB:92z.}NYuCLnQ
4<"=Z7h&PA1\| "nA'A

Header Hexadecimal Representation:
ff d8 ff e0 00 10 4a 46 49 46 00 01 01 00 00 01

File Type: JPEG Image
File Size: 7 KB
Details: This is a JPEG image file.
Compression: Baseline DCT, Huffman coding
Color Space: YCbCr
```