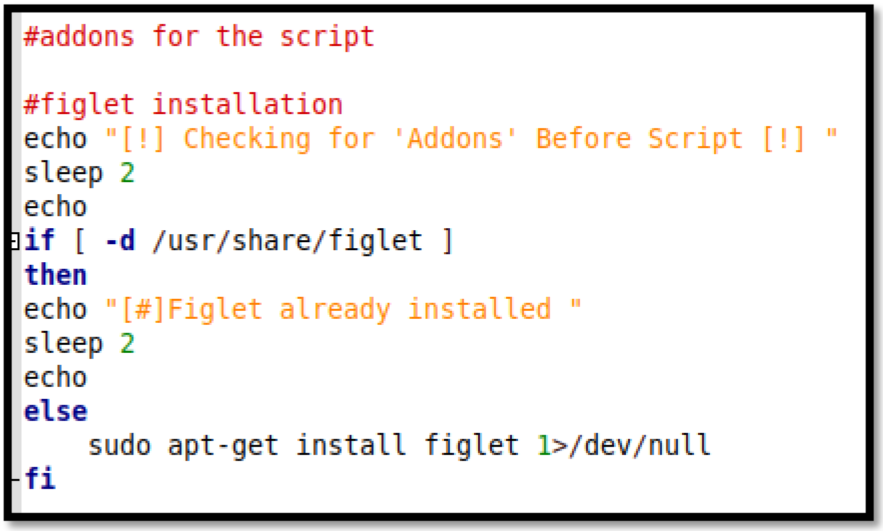
# Project Windows Forensics - Ofir Halfin

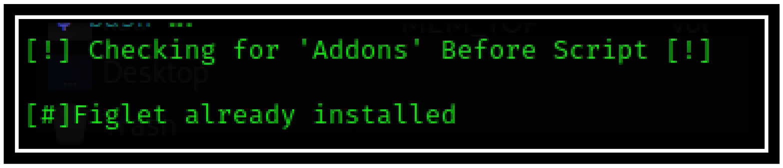
Before the script, start installing Figlet and then start the script

## input



```
#addons for the script

#figlet installation
echo "[!] Checking for 'Addons' Before Script [!] "
sleep 2
echo
if [ -d /usr/share/figlet ]
then
echo "[#]Figlet already installed "
sleep 2
echo
else
    sudo apt-get install figlet 1>/dev/null
fi
```

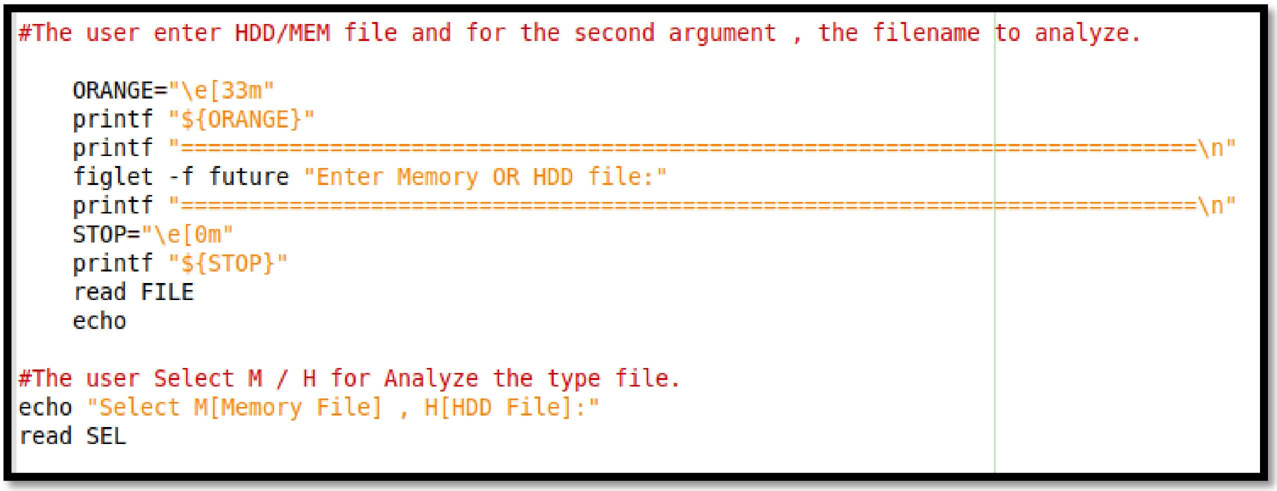
## output



```
[!] Checking for 'Addons' Before Script [!]

[#]Figlet already installed
```

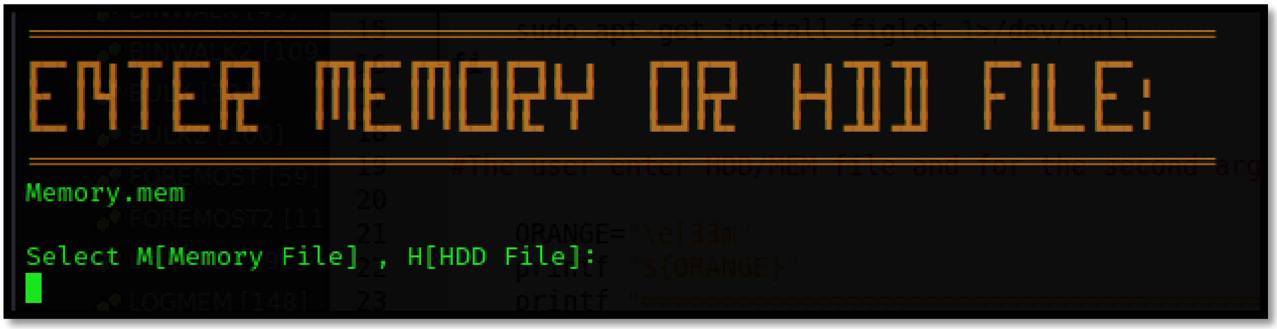
## The user enters HDD/MEM file and for the second argument, the filename to analyze it

## input



```
#The user enter HDD/MEM file and for the second argument , the filename to analyze.

    ORANGE="\e[33m"
    printf "${ORANGE}"
    printf "===========================================================================\n"
    figlet -f future "Enter Memory OR HDD file:"
    printf "===========================================================================\n"
    STOP="\e[0m"
    printf "${STOP}"
    read FILE
    echo

#The user Select M / H for Analyze the type file.
echo "Select M[Memory File] , H[HDD File]:"
read SEL
```

## output



```
===========================================================================
ENTER MEMORY OR HDD FILE:
===========================================================================
Memory.mem

Select M[Memory File] , H[HDD File]:
```

## Functions for Memory file (Bulk, Binwalk, Foremost, Strings, Vol):

Bulk-Extractor- bulk-extractor is a C++ program that scans a disk image, a file, or a directory of files and extracts useful information without parsing the file system or file system structures. The results are stored in feature files that can be easily inspected, parsed, or processed with automated tools. Bulk-extractor also creates histograms of features that it finds, as features that are more common tend to be more important.

Binwalk- Binwalk is a tool for searching a given binary image for embedded files and executable code. Specifically, it is designed for identifying files and code embedded inside of firmware images. Binwalk uses the libmagic library, so it is compatible with magic signatures created for the Unix file utility.

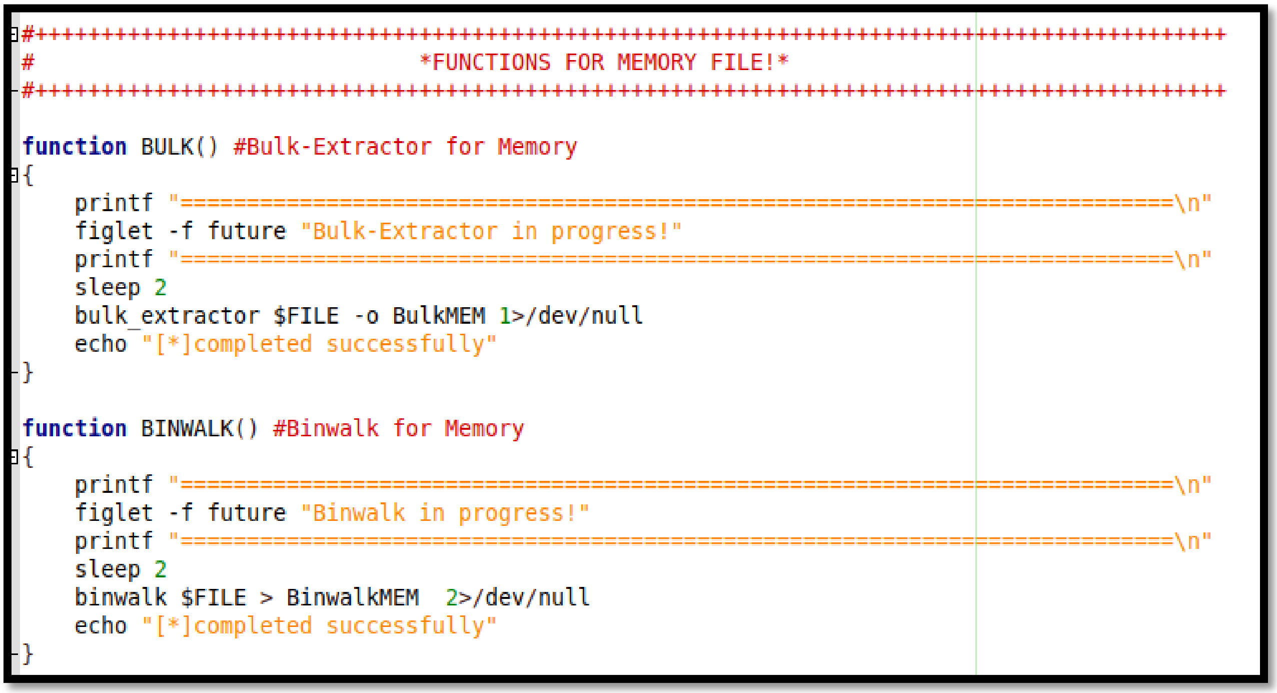
Foremost- Foremost is a forensic program to recover lost files based on their headers, footers, and internal data structures.

Foremost can work on image files, such as those generated by dd, Safeback, Encase, etc, or directly on a drive.

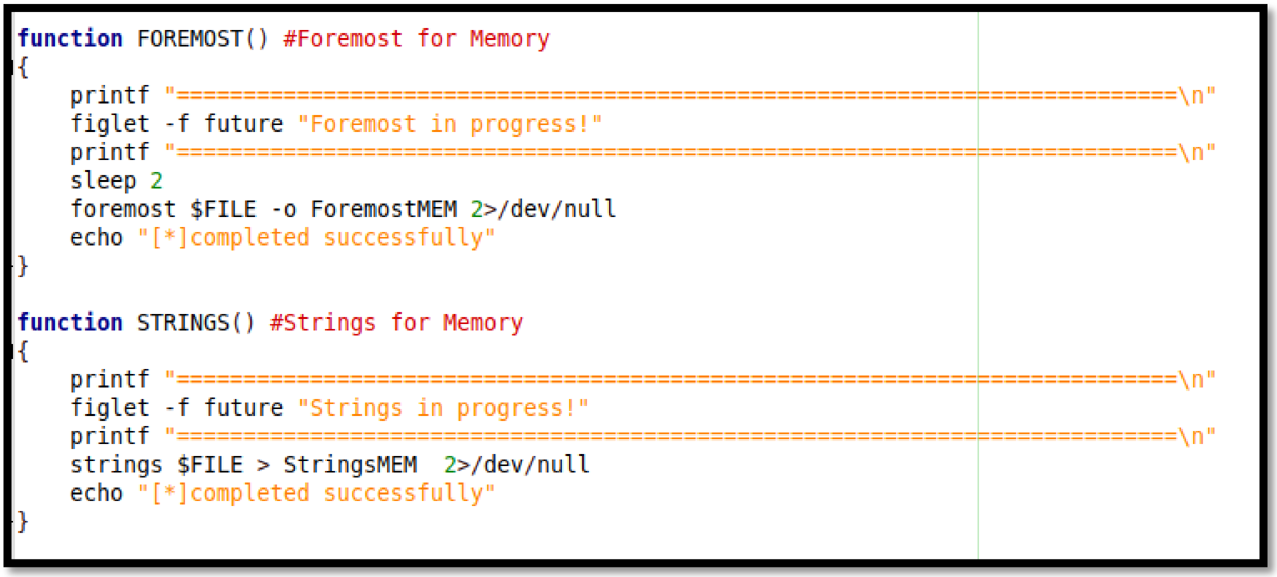
Strings- Linux strings command is used to return the string characters into files. It primarily focuses on determining the contents of and extracting text from the binary files (non-text file). It is a complex task for a human to find out text from an executable file.

Volatility- Volatility can be used during an investigation to link artifacts from the device, network, file system, and registry to ascertain the list of all running processes, active and closed network connections, running Windows command prompts, screenshots, and clipboard contents that ran within the timeframe of the incident.

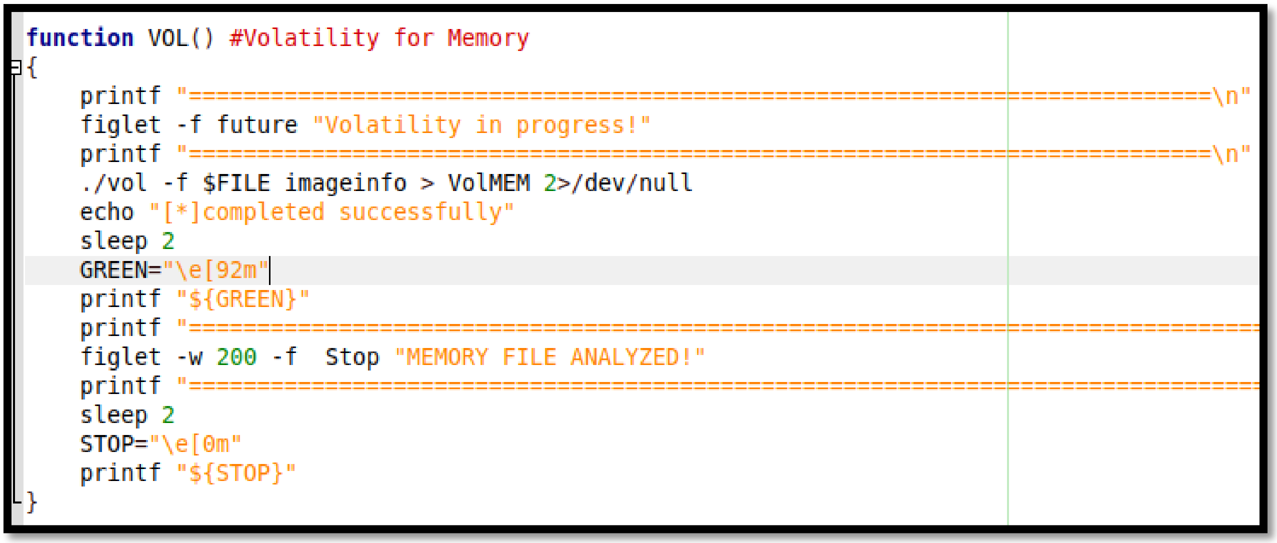
# input



```
#+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
#                             *FUNCTIONS FOR MEMORY FILE!*
#+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++

function BULK() #Bulk-Extractor for Memory
{
    printf "============================================================================\n"
    figlet -f future "Bulk-Extractor in progress!"
    printf "============================================================================\n"
    sleep 2
    bulk_extractor $FILE -o BulkMEM 1>/dev/null
    echo "[*]completed successfully"
}

function BINWALK() #Binwalk for Memory
{
    printf "============================================================================\n"
    figlet -f future "Binwalk in progress!"
    printf "============================================================================\n"
    sleep 2
    binwalk $FILE > BinwalkMEM  2>/dev/null
    echo "[*]completed successfully"
}
```

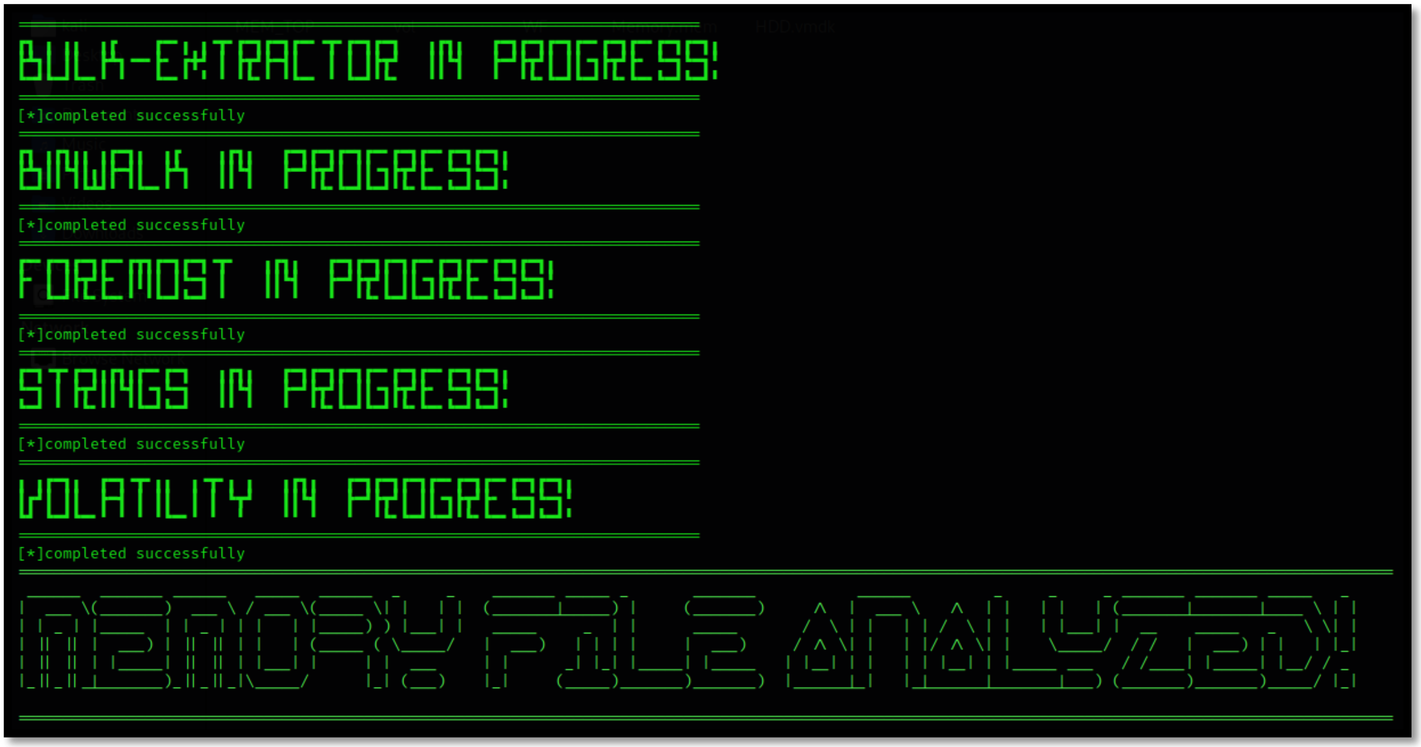


```
function FOREMOST() #Foremost for Memory
{
    printf "============================================================================\n"
    figlet -f future "Foremost in progress!"
    printf "============================================================================\n"
    sleep 2
    foremost $FILE -o ForemostMEM 2>/dev/null
    echo "[*]completed successfully"
}

function STRINGS() #Strings for Memory
{
    printf "============================================================================\n"
    figlet -f future "Strings in progress!"
    printf "============================================================================\n"
    strings $FILE > StringsMEM  2>/dev/null
    echo "[*]completed successfully"
}
```



```
function VOL() #Volatility for Memory
{
    printf "============================================================================\n"
    figlet -f future "Volatility in progress!"
    printf "============================================================================\n"
    ./vol -f $FILE imageinfo > VolMEM 2>/dev/null
    echo "[*]completed successfully"
    sleep 2
    GREEN="\e[92m"
    printf "${GREEN}"
    printf "===============================================================================
    figlet -w 200 -f  Stop "MEMORY FILE ANALYZED!"
    printf "===============================================================================
    sleep 2
    STOP="\e[0m"
    printf "${STOP}"
}
```

output

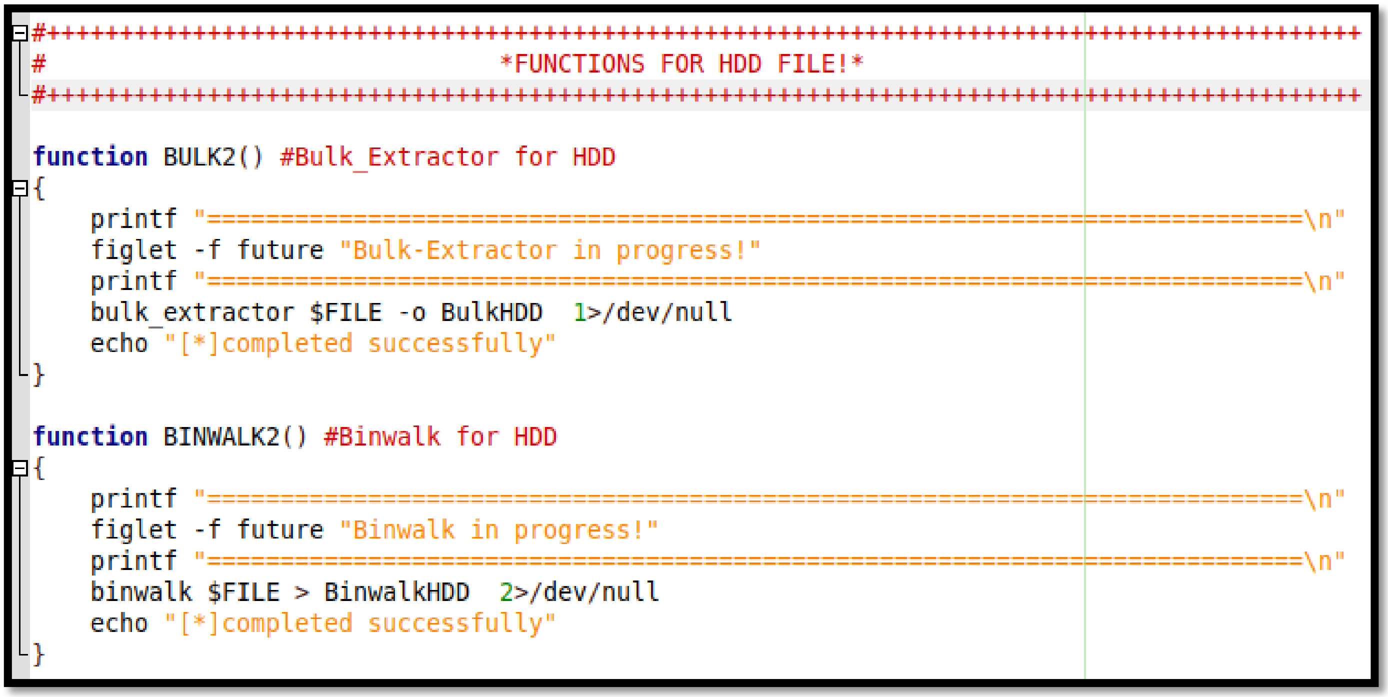


```
BULK-EXTRACTOR IN PROGRESS!
[*]completed successfully
BINWALK IN PROGRESS!
[*]completed successfully
FOREMOST IN PROGRESS!
[*]completed successfully
STRINGS IN PROGRESS!
[*]completed successfully
VOLATILITY IN PROGRESS!
[*]completed successfully
MEMORY FILE ANALYZED!
```

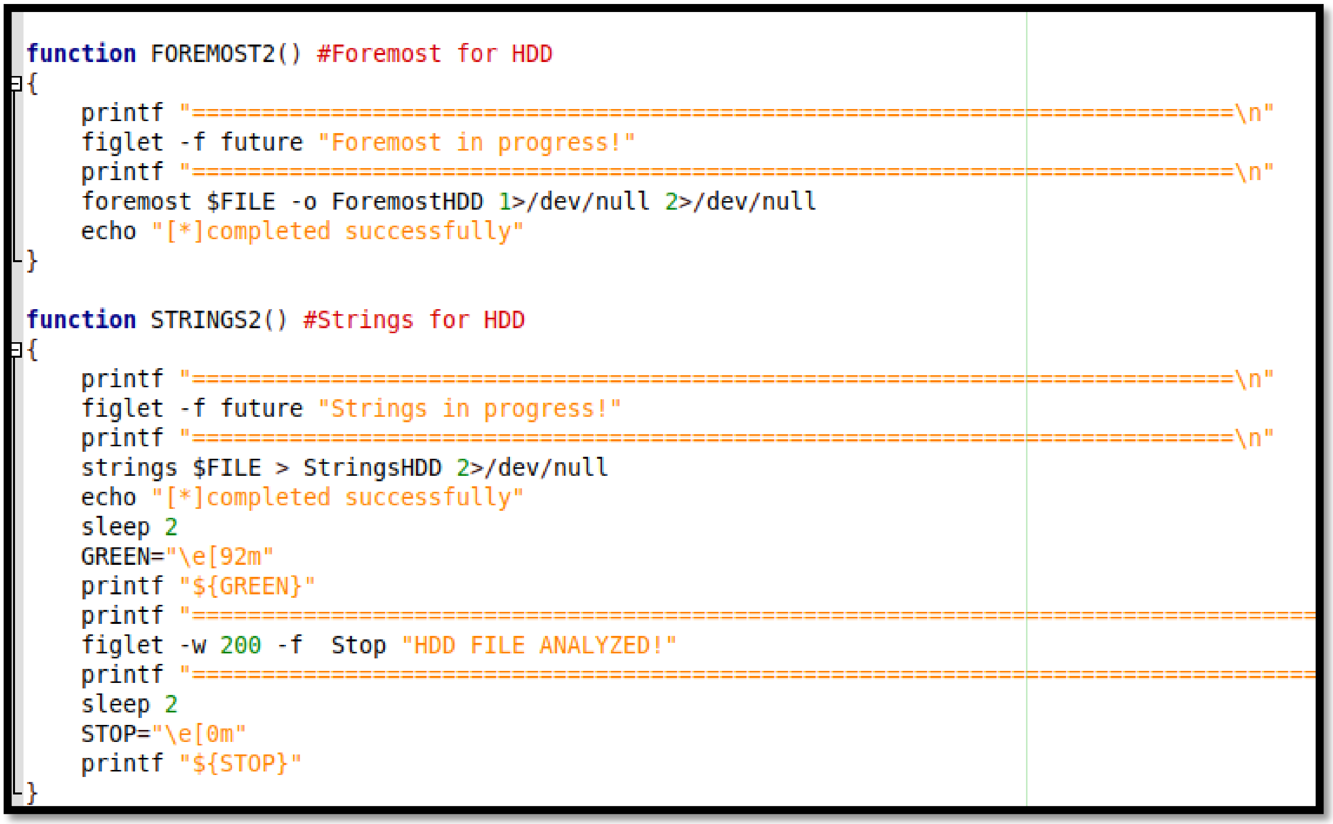
## Functions for HDD file (Bulk, Binwalk, Foremost, Strings):

The meaning of each one is listed above!

input

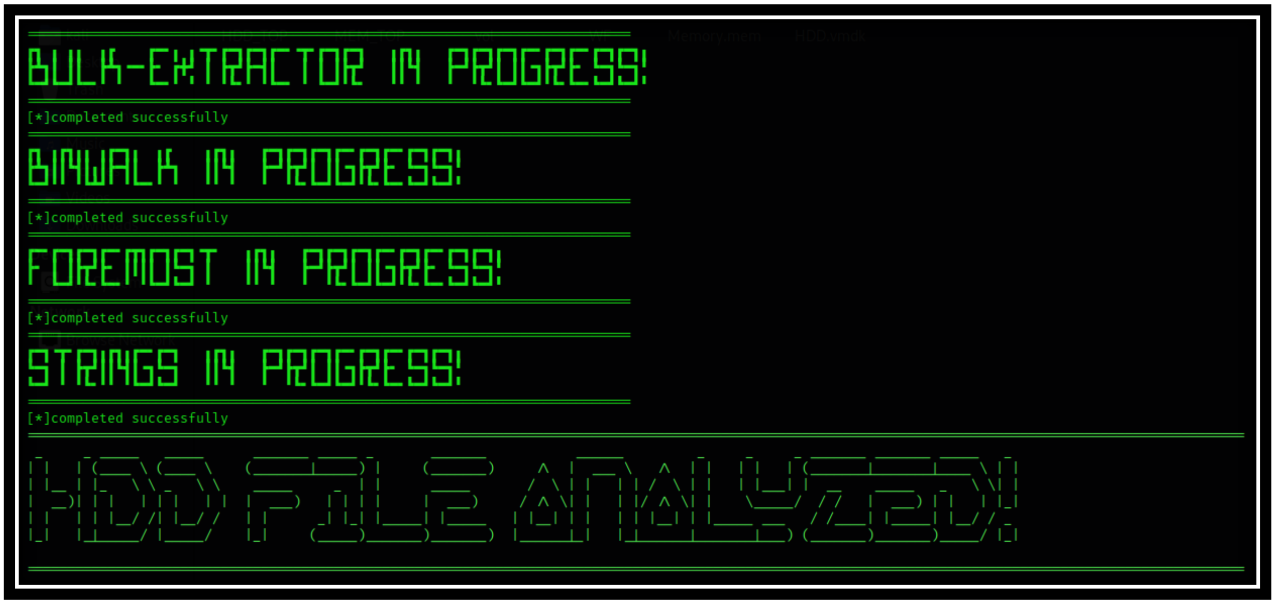


```
#++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
#                               *FUNCTIONS FOR HDD FILE!*
#++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++

function BULK2() #Bulk_Extractor for HDD
{
    printf "===========================================================================\n"
    figlet -f future "Bulk-Extractor in progress!"
    printf "===========================================================================\n"
    bulk_extractor $FILE -o BulkHDD  1>/dev/null
    echo "[*]completed successfully"
}

function BINWALK2() #Binwalk for HDD
{
    printf "===========================================================================\n"
    figlet -f future "Binwalk in progress!"
    printf "===========================================================================\n"
    binwalk $FILE > BinwalkHDD  2>/dev/null
    echo "[*]completed successfully"
}
```



```
function FOREMOST2() #Foremost for HDD
{
    printf "===========================================================================\n"
    figlet -f future "Foremost in progress!"
    printf "===========================================================================\n"
    foremost $FILE -o ForemostHDD 1>/dev/null 2>/dev/null
    echo "[*]completed successfully"
}

function STRINGS2() #Strings for HDD
{
    printf "===========================================================================\n"
    figlet -f future "Strings in progress!"
    printf "===========================================================================\n"
    strings $FILE > StringsHDD 2>/dev/null
    echo "[*]completed successfully"
    sleep 2
    GREEN="\e[92m"
    printf "${GREEN}"
    printf "=================================================================================
    figlet -w 200 -f  Stop "HDD FILE ANALYZED!"
    printf "=================================================================================
    sleep 2
    STOP="\e[0m"
    printf "${STOP}"
}
```

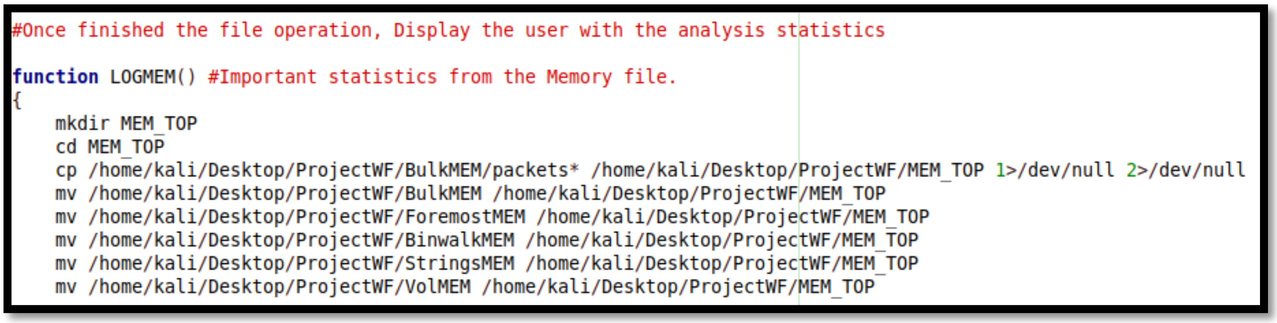
# output



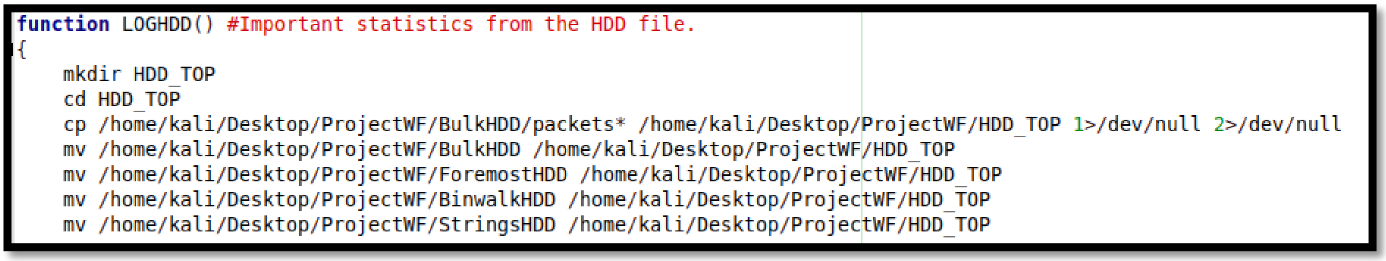
```
============================================
BULK-EXTRACTOR IN PROGRESS!
============================================
[*]completed successfully
============================================
BINWALK IN PROGRESS!
============================================
[*]completed successfully
============================================
FOREMOST IN PROGRESS!
============================================
[*]completed successfully
============================================
STRINGS IN PROGRESS!
============================================
[*]completed successfully
==========================================================================================
HDD FILE ANALYZED!
==========================================================================================
```

After analyzing the file, the extracted files will be moved to a special folder.

# input



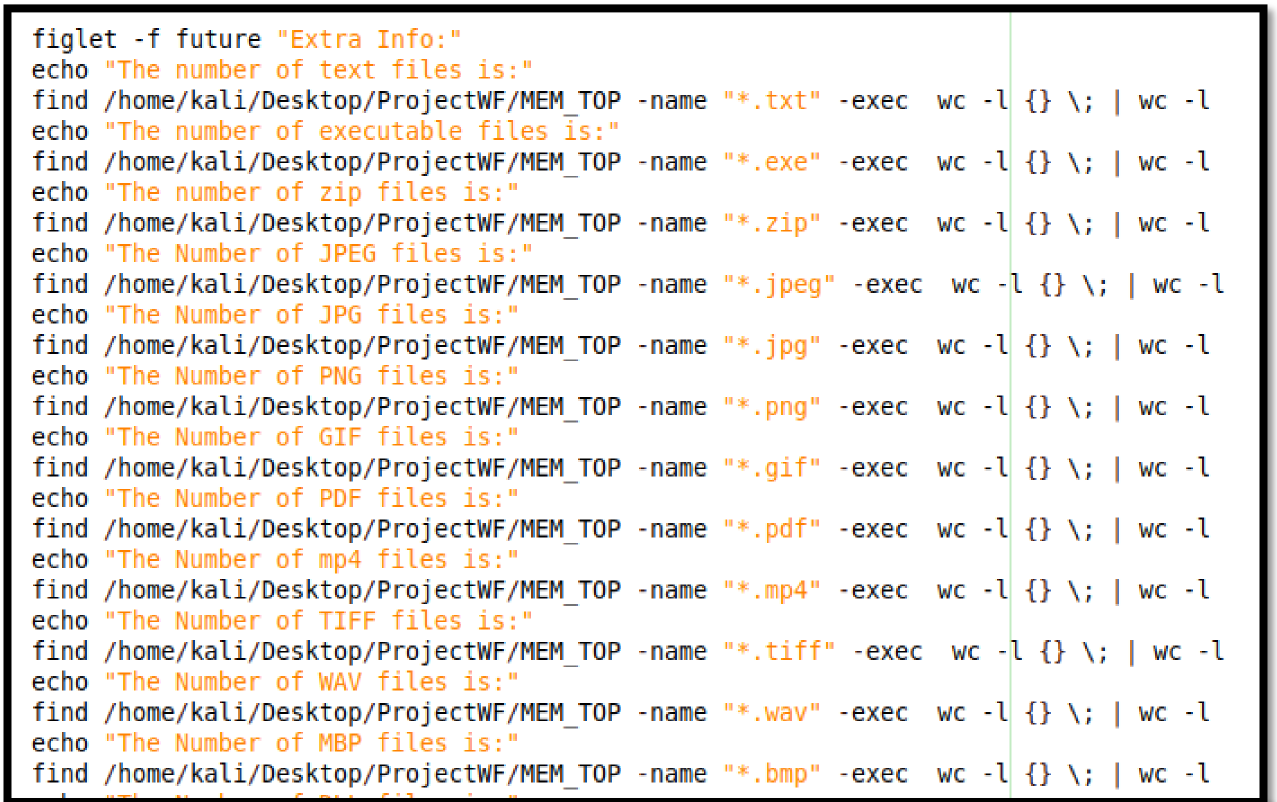
```
#Once finished the file operation, Display the user with the analysis statistics

function LOGMEM() #Important statistics from the Memory file.
{
    mkdir MEM_TOP
    cd MEM_TOP
    cp /home/kali/Desktop/ProjectWF/BulkMEM/packets* /home/kali/Desktop/ProjectWF/MEM_TOP 1>/dev/null 2>/dev/null
    mv /home/kali/Desktop/ProjectWF/BulkMEM /home/kali/Desktop/ProjectWF/MEM_TOP
    mv /home/kali/Desktop/ProjectWF/ForemostMEM /home/kali/Desktop/ProjectWF/MEM_TOP
    mv /home/kali/Desktop/ProjectWF/BinwalkMEM /home/kali/Desktop/ProjectWF/MEM_TOP
    mv /home/kali/Desktop/ProjectWF/StringsMEM /home/kali/Desktop/ProjectWF/MEM_TOP
    mv /home/kali/Desktop/ProjectWF/VolMEM /home/kali/Desktop/ProjectWF/MEM_TOP
```



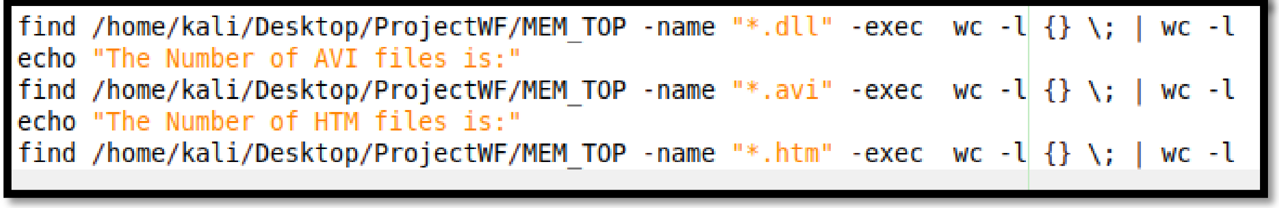
```
function LOGHDD() #Important statistics from the HDD file.
{
    mkdir HDD_TOP
    cd HDD_TOP
    cp /home/kali/Desktop/ProjectWF/BulkHDD/packets* /home/kali/Desktop/ProjectWF/HDD_TOP 1>/dev/null 2>/dev/null
    mv /home/kali/Desktop/ProjectWF/BulkHDD /home/kali/Desktop/ProjectWF/HDD_TOP
    mv /home/kali/Desktop/ProjectWF/ForemostHDD /home/kali/Desktop/ProjectWF/HDD_TOP
    mv /home/kali/Desktop/ProjectWF/BinwalkHDD /home/kali/Desktop/ProjectWF/HDD_TOP
    mv /home/kali/Desktop/ProjectWF/StringsHDD /home/kali/Desktop/ProjectWF/HDD_TOP
```

After analyzing the file, the important files in the extracted file will be counted and printed.

# Input for Memory

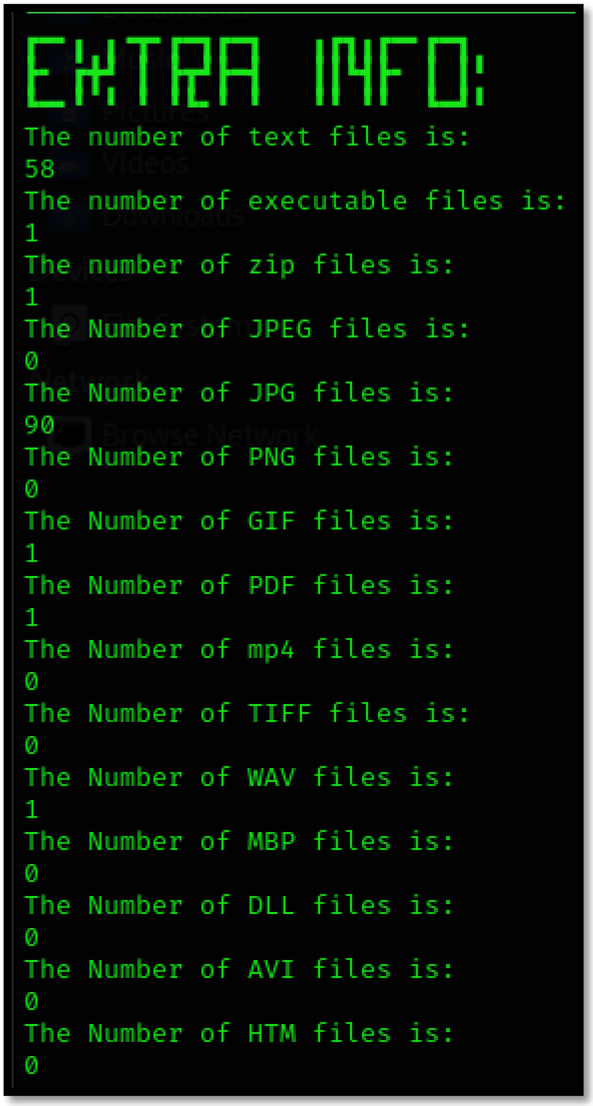


```
figlet -f future "Extra Info:"
echo "The number of text files is:"
find /home/kali/Desktop/ProjectWF/MEM_TOP -name "*.txt" -exec  wc -l {} \; | wc -l
echo "The number of executable files is:"
find /home/kali/Desktop/ProjectWF/MEM_TOP -name "*.exe" -exec  wc -l {} \; | wc -l
echo "The number of zip files is:"
find /home/kali/Desktop/ProjectWF/MEM_TOP -name "*.zip" -exec  wc -l {} \; | wc -l
echo "The Number of JPEG files is:"
find /home/kali/Desktop/ProjectWF/MEM_TOP -name "*.jpeg" -exec  wc -l {} \; | wc -l
echo "The Number of JPG files is:"
find /home/kali/Desktop/ProjectWF/MEM_TOP -name "*.jpg" -exec  wc -l {} \; | wc -l
echo "The Number of PNG files is:"
find /home/kali/Desktop/ProjectWF/MEM_TOP -name "*.png" -exec  wc -l {} \; | wc -l
echo "The Number of GIF files is:"
find /home/kali/Desktop/ProjectWF/MEM_TOP -name "*.gif" -exec  wc -l {} \; | wc -l
echo "The Number of PDF files is:"
find /home/kali/Desktop/ProjectWF/MEM_TOP -name "*.pdf" -exec  wc -l {} \; | wc -l
echo "The Number of mp4 files is:"
find /home/kali/Desktop/ProjectWF/MEM_TOP -name "*.mp4" -exec  wc -l {} \; | wc -l
echo "The Number of TIFF files is:"
find /home/kali/Desktop/ProjectWF/MEM_TOP -name "*.tiff" -exec  wc -l {} \; | wc -l
echo "The Number of WAV files is:"
find /home/kali/Desktop/ProjectWF/MEM_TOP -name "*.wav" -exec  wc -l {} \; | wc -l
echo "The Number of MBP files is:"
find /home/kali/Desktop/ProjectWF/MEM_TOP -name "*.bmp" -exec  wc -l {} \; | wc -l
echo "The Number of DLL files is:"
```



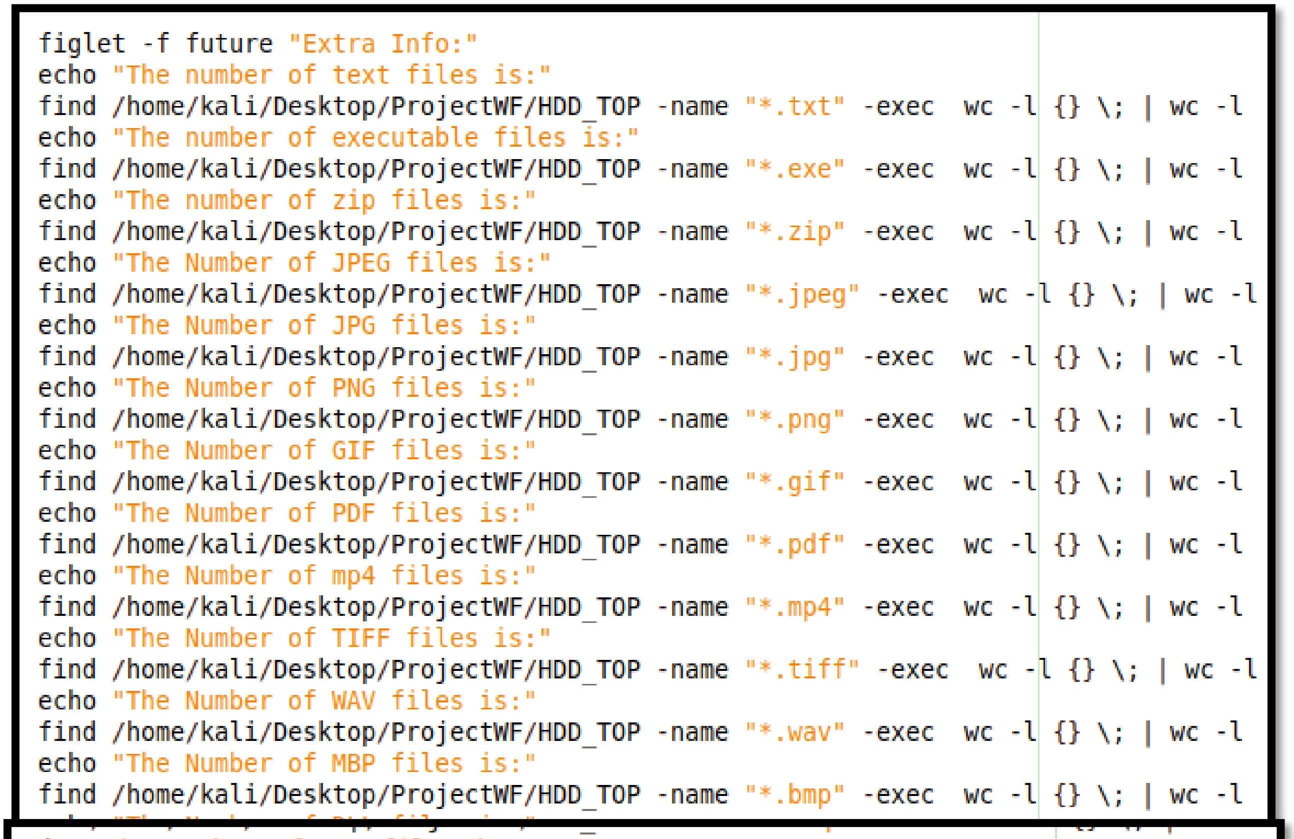
```
find /home/kali/Desktop/ProjectWF/MEM_TOP -name "*.dll" -exec  wc -l {} \; | wc -l
echo "The Number of AVI files is:"
find /home/kali/Desktop/ProjectWF/MEM_TOP -name "*.avi" -exec  wc -l {} \; | wc -l
echo "The Number of HTM files is:"
find /home/kali/Desktop/ProjectWF/MEM_TOP -name "*.htm" -exec  wc -l {} \; | wc -l
```

# Output for Memory

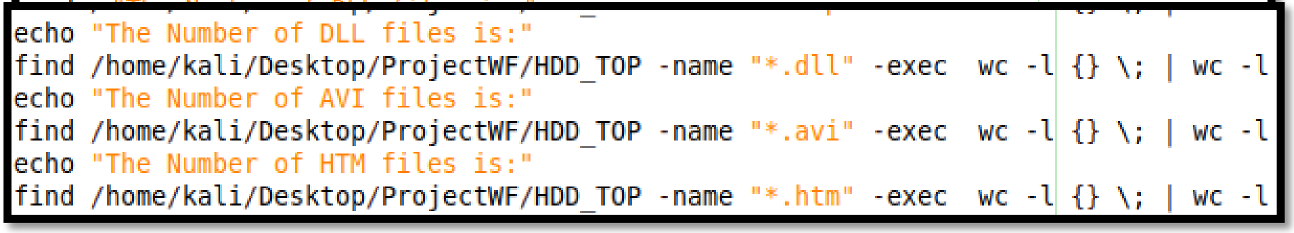


```
EXTRA INFO:
The number of text files is:
58
The number of executable files is:
1
The number of zip files is:
1
The Number of JPEG files is:
0
The Number of JPG files is:
90
The Number of PNG files is:
0
The Number of GIF files is:
1
The Number of PDF files is:
1
The Number of mp4 files is:
0
The Number of TIFF files is:
0
The Number of WAV files is:
1
The Number of MBP files is:
0
The Number of DLL files is:
0
The Number of AVI files is:
0
The Number of HTM files is:
0
```

# Input for HDD

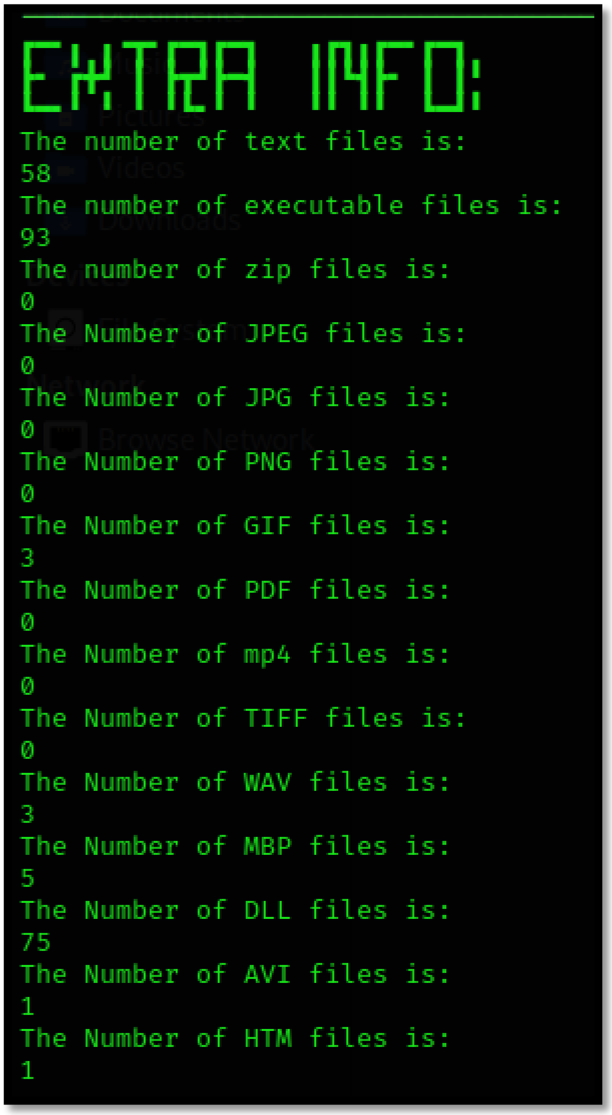


```
figlet -f future "Extra Info:"
echo "The number of text files is:"
find /home/kali/Desktop/ProjectWF/HDD_TOP -name "*.txt" -exec  wc -l {} \; | wc -l
echo "The number of executable files is:"
find /home/kali/Desktop/ProjectWF/HDD_TOP -name "*.exe" -exec  wc -l {} \; | wc -l
echo "The number of zip files is:"
find /home/kali/Desktop/ProjectWF/HDD_TOP -name "*.zip" -exec  wc -l {} \; | wc -l
echo "The Number of JPEG files is:"
find /home/kali/Desktop/ProjectWF/HDD_TOP -name "*.jpeg" -exec  wc -l {} \; | wc -l
echo "The Number of JPG files is:"
find /home/kali/Desktop/ProjectWF/HDD_TOP -name "*.jpg" -exec  wc -l {} \; | wc -l
echo "The Number of PNG files is:"
find /home/kali/Desktop/ProjectWF/HDD_TOP -name "*.png" -exec  wc -l {} \; | wc -l
echo "The Number of GIF files is:"
find /home/kali/Desktop/ProjectWF/HDD_TOP -name "*.gif" -exec  wc -l {} \; | wc -l
echo "The Number of PDF files is:"
find /home/kali/Desktop/ProjectWF/HDD_TOP -name "*.pdf" -exec  wc -l {} \; | wc -l
echo "The Number of mp4 files is:"
find /home/kali/Desktop/ProjectWF/HDD_TOP -name "*.mp4" -exec  wc -l {} \; | wc -l
echo "The Number of TIFF files is:"
find /home/kali/Desktop/ProjectWF/HDD_TOP -name "*.tiff" -exec  wc -l {} \; | wc -l
echo "The Number of WAV files is:"
find /home/kali/Desktop/ProjectWF/HDD_TOP -name "*.wav" -exec  wc -l {} \; | wc -l
echo "The Number of MBP files is:"
find /home/kali/Desktop/ProjectWF/HDD_TOP -name "*.bmp" -exec  wc -l {} \; | wc -l
```



```
echo "The Number of DLL files is:"
find /home/kali/Desktop/ProjectWF/HDD_TOP -name "*.dll" -exec  wc -l {} \; | wc -l
echo "The Number of AVI files is:"
find /home/kali/Desktop/ProjectWF/HDD_TOP -name "*.avi" -exec  wc -l {} \; | wc -l
echo "The Number of HTM files is:"
find /home/kali/Desktop/ProjectWF/HDD_TOP -name "*.htm" -exec  wc -l {} \; | wc -l
```

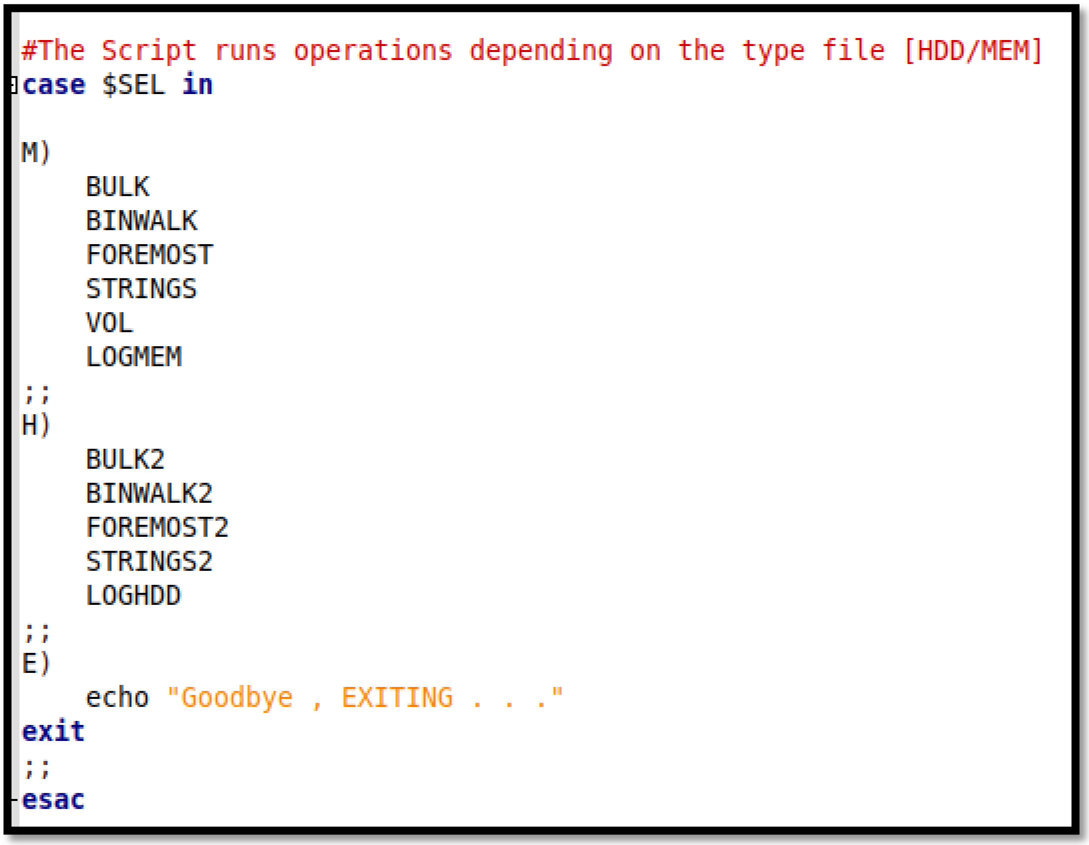
# Output for HDD



```
EXTRA INFO:
The number of text files is:
58
The number of executable files is:
93
The number of zip files is:
0
The Number of JPEG files is:
0
The Number of JPG files is:
0
The Number of PNG files is:
0
The Number of GIF files is:
3
The Number of PDF files is:
0
The Number of mp4 files is:
0
The Number of TIFF files is:
0
The Number of WAV files is:
3
The Number of MBP files is:
5
The Number of DLL files is:
75
The Number of AVI files is:
1
The Number of HTM files is:
1
```

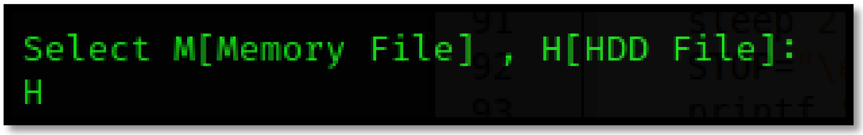
## The script runs operations depending on the type file [HDD/MEM]

# input



```
#The Script runs operations depending on the type file [HDD/MEM]
case $SEL in

M)
    BULK
    BINWALK
    FOREMOST
    STRINGS
    VOL
    LOGMEM
;;
H)
    BULK2
    BINWALK2
    FOREMOST2
    STRINGS2
    LOGHDD
;;
E)
    echo "Goodbye , EXITING . . ."
exit
;;
esac
```

# Output



```
Select M[Memory File] , H[HDD File]:
H
```