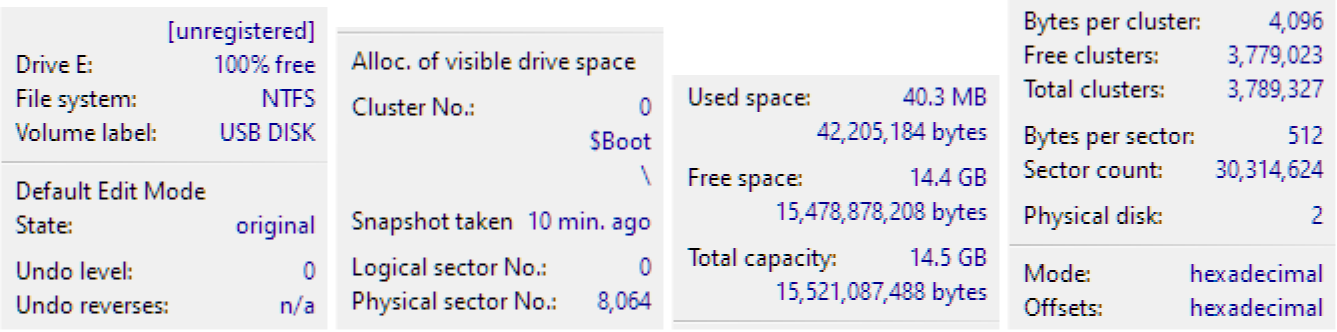
Charles Wallis
2/12/2023
CS 4398.001 Digital Forensics
Dr. Gupta

1. Show my device information



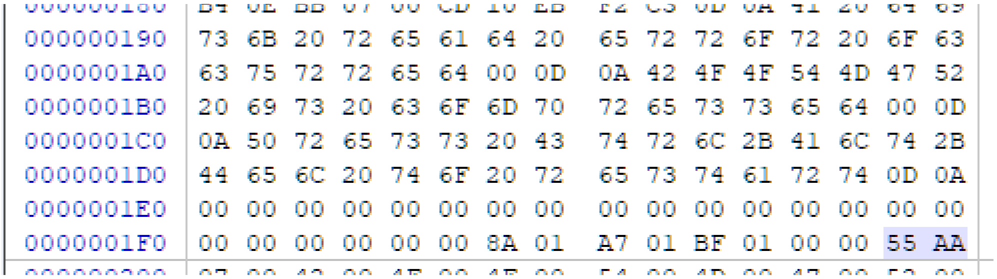
```
                [unregistered]
Drive E:             100% free
File system:              NTFS
Volume label:         USB DISK

Default Edit Mode
State:                original

Undo level:                  0
Undo reverses:             n/a
```

```
Alloc. of visible drive space
Cluster No.:                 0
                         $Boot
                             \
Snapshot taken   10 min. ago

Logical sector No.:          0
Physical sector No.:     8,064
```

```
Used space:            40.3 MB
              42,205,184 bytes

Free space:            14.4 GB
          15,478,878,208 bytes
Total capacity:        14.5 GB
          15,521,087,488 bytes
```

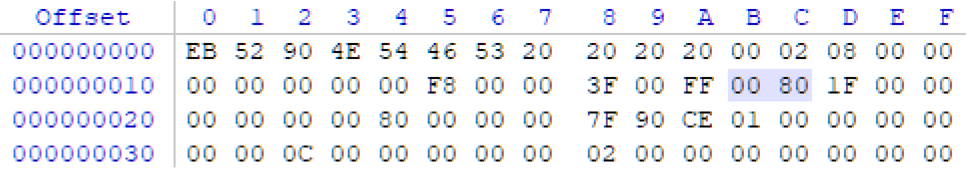
```
Bytes per cluster:       4,096
Free clusters:       3,779,023
Total clusters:      3,789,327

Bytes per sector:          512
Sector count:       30,314,624

Physical disk:               2

Mode:              hexadecimal
Offsets:           hexadecimal
```

2. Highlight MBR signature



```
000000180  B4 0E BB 07 00 CD 10 EB  F2 C3 0D 0A 41 20 64 69
000000190  73 6B 20 72 65 61 64 20  65 72 72 6F 72 20 6F 63
0000001A0  63 75 72 72 65 64 00 0D  0A 42 4F 4F 54 4D 47 52
0000001B0  20 69 73 20 63 6F 6D 70  72 65 73 73 65 64 00 0D
0000001C0  0A 50 72 65 73 73 20 43  74 72 6C 2B 41 6C 74 2B
0000001D0  44 65 6C 20 74 6F 20 72  65 73 74 61 72 74 0D 0A
0000001E0  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00
0000001F0  00 00 00 00 00 00 8A 01  A7 01 BF 01 00 00 55 AA
000000200  07 00 42 00 4F 00 4F 00  54 00 4D 00 47 00 52 00
```

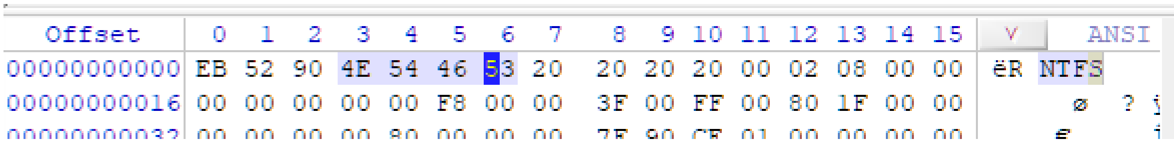
3. Highlight LBA of my partition



```
  Offset    0  1  2  3  4  5  6  7   8  9  A  B  C  D  E  F
000000000  EB 52 90 4E 54 46 53 20  20 20 20 00 02 08 00 00
000000010  00 00 00 00 00 F8 00 00  3F 00 FF 00 80 1F 00 00
000000020  00 00 00 00 80 00 00 00  7F 90 CE 01 00 00 00 00
000000030  00 00 0C 00 00 00 00 00  02 00 00 00 00 00 00 00
```

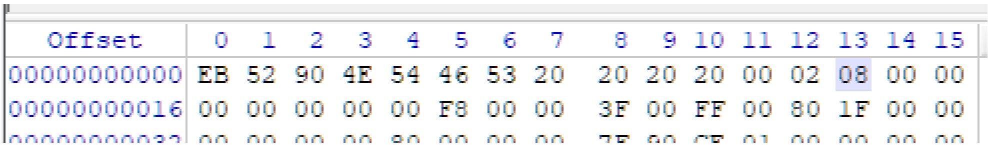
4. From the LBA value, calculate the start of my partition
To do this, I need to multiply the LBA value by sector size, which is 512 bytes (as shown in part 1)
The LBA value is shown above from part 3, 08 00, or a value of 2048.
The partition starts at sector 2048, or 2048*512 = 1048576 bytes offset (0x100000)

5. The Superblock Magic Signature is NTFS in Hex, 4E 54 46 53.



```
   Offset     0  1  2  3  4  5  6  7   8  9 10 11 12 13 14 15   V    ANSI
00000000000  EB 52 90 4E 54 46 53 20  20 20 20 00 02 08 00 00   ëR NTFS
00000000016  00 00 00 00 00 F8 00 00  3F 00 FF 00 80 1F 00 00        ø ? ÿ
00000000032  00 00 00 00 80 00 00 00  7F 90 CE 01 00 00 00 00       €    Î
```

6. 0x0D indicates sectors per cluster, and there are 8 sectors per cluster



```
   Offset     0  1  2  3  4  5  6  7   8  9 10 11 12 13 14 15
00000000000  EB 52 90 4E 54 46 53 20  20 20 20 00 02 08 00 00
00000000016  00 00 00 00 00 F8 00 00  3F 00 FF 00 80 1F 00 00
00000000032  00 00 00 00 80 00 00 00  7F 90 CE 01 00 00 00 00
```

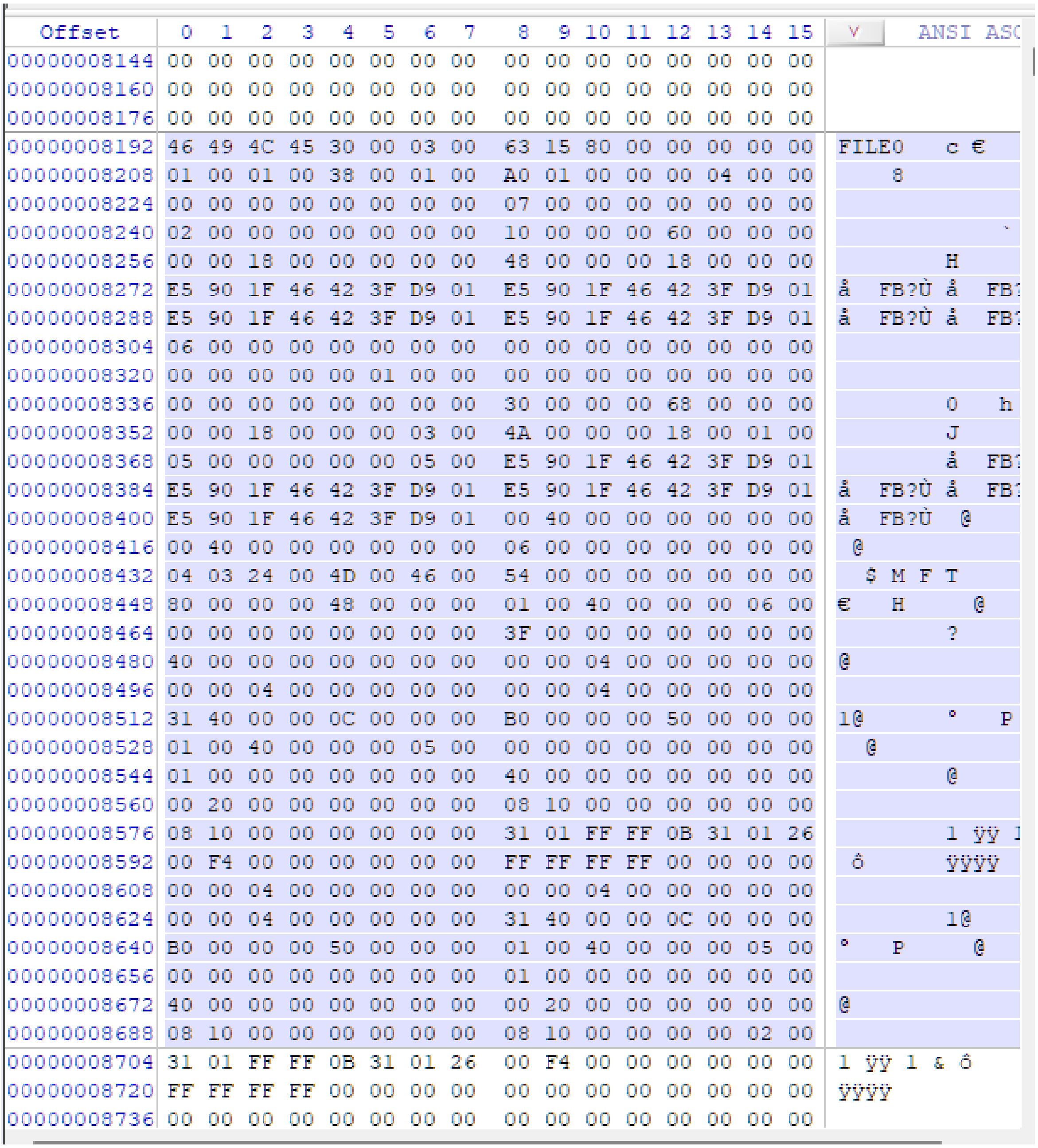
The block size is 512/8 since there are 8 sectors and each sector is 512 bytes (shown in part 1)

Each sector(block) is 64 bytes.

7. Number of blocks(sectors) per group (cluster) can be calculated using the information shown in part 1, since it tells us that there are 4096 bytes per cluster and each block is 512 bytes. 4096/512 = 8 blocks.

8. Since the superblock is in the first block it was in block #1. Each group has blocks consisting: super block, group descriptors, data block bitmap, inode bitmap, inode table, and a data block. The number of blocks are always an integral power of 2

9. To get to block group 3, I need to multiply 2x4096 to jump to the location at offset 0x8192



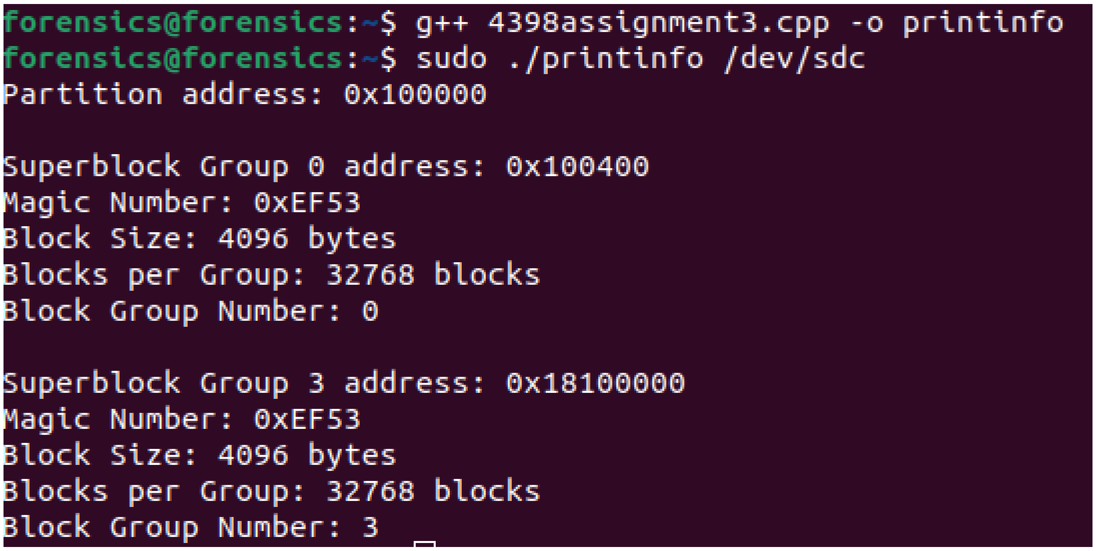
```
   Offset     0  1  2  3  4  5  6  7   8  9 10 11 12 13 14 15   ANSI ASC
00000008144  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00
00000008160  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00
00000008176  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00
00000008192  46 49 4C 45 30 00 03 00  63 15 80 00 00 00 00 00   FILE0   c €
00000008208  01 00 01 00 38 00 01 00  A0 01 00 00 00 04 00 00       8
00000008224  00 00 00 00 00 00 00 00  07 00 00 00 00 00 00 00
00000008240  02 00 00 00 00 00 00 00  10 00 00 00 60 00 00 00               `
00000008256  00 00 18 00 00 00 00 00  48 00 00 00 18 00 00 00           H
00000008272  E5 90 1F 46 42 3F D9 01  E5 90 1F 46 42 3F D9 01   å  FB?Ù å  FB
00000008288  E5 90 1F 46 42 3F D9 01  E5 90 1F 46 42 3F D9 01   å  FB?Ù å  FB
00000008304  06 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00
00000008320  00 00 00 00 00 01 00 00  00 00 00 00 00 00 00 00
00000008336  00 00 00 00 00 00 00 00  30 00 00 00 68 00 00 00           0   h
00000008352  00 00 18 00 00 00 03 00  4A 00 00 00 18 00 01 00           J
00000008368  05 00 00 00 00 00 05 00  E5 90 1F 46 42 3F D9 01           å  FB
00000008384  E5 90 1F 46 42 3F D9 01  E5 90 1F 46 42 3F D9 01   å  FB?Ù å  FB
00000008400  E5 90 1F 46 42 3F D9 01  00 40 00 00 00 00 00 00   å  FB?Ù  @
00000008416  00 40 00 00 00 00 00 00  06 00 00 00 00 00 00 00    @
00000008432  04 03 24 00 4D 00 46 00  54 00 00 00 00 00 00 00     $ M F T
00000008448  80 00 00 00 48 00 00 00  01 00 40 00 00 00 06 00   €   H     @
00000008464  00 00 00 00 00 00 00 00  3F 00 00 00 00 00 00 00           ?
00000008480  40 00 00 00 00 00 00 00  00 00 04 00 00 00 00 00   @
00000008496  00 00 04 00 00 00 00 00  00 00 04 00 00 00 00 00
00000008512  31 40 00 00 0C 00 00 00  B0 00 00 00 50 00 00 00   1@      °   P
00000008528  01 00 40 00 00 00 05 00  00 00 00 00 00 00 00 00     @
00000008544  01 00 00 00 00 00 00 00  40 00 00 00 00 00 00 00           @
00000008560  00 20 00 00 00 00 00 00  08 10 00 00 00 00 00 00
00000008576  08 10 00 00 00 00 00 00  31 01 FF FF 0B 31 01 26           1 ÿÿ 1
00000008592  00 F4 00 00 00 00 00 00  FF FF FF FF 00 00 00 00    ô      ÿÿÿÿ
00000008608  00 00 04 00 00 00 00 00  00 00 04 00 00 00 00 00
00000008624  00 00 04 00 00 00 00 00  31 40 00 00 0C 00 00 00           1@
00000008640  B0 00 00 00 50 00 00 00  01 00 40 00 00 00 05 00   °   P     @
00000008656  00 00 00 00 00 00 00 00  01 00 00 00 00 00 00 00
00000008672  40 00 00 00 00 00 00 00  00 20 00 00 00 00 00 00   @
00000008688  08 10 00 00 00 00 00 00  08 10 00 00 00 00 02 00
00000008704  31 01 FF FF 0B 31 01 26  00 F4 00 00 00 00 00 00   1 ÿÿ 1 & ô
00000008720  FF FF FF FF 00 00 00 00  00 00 00 00 00 00 00 00   ÿÿÿÿ
00000008736  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00
```

In a block group, the first block will always be the superblock, so here is the first block of block group 3.

Now, I will automate this through writing a c++ program, on linux.

The source code is attached on github,
https://github.com/charlestw127/Digital-Forensics/blob/main/Hexedit%20Diagnose.cpp

Here is the output of the program:



```
forensics@forensics:~$ g++ 4398assignment3.cpp -o printinfo
forensics@forensics:~$ sudo ./printinfo /dev/sdc
Partition address: 0x100000

Superblock Group 0 address: 0x100400
Magic Number: 0xEF53
Block Size: 4096 bytes
Blocks per Group: 32768 blocks
Block Group Number: 0

Superblock Group 3 address: 0x18100000
Magic Number: 0xEF53
Block Size: 4096 bytes
Blocks per Group: 32768 blocks
Block Group Number: 3
```