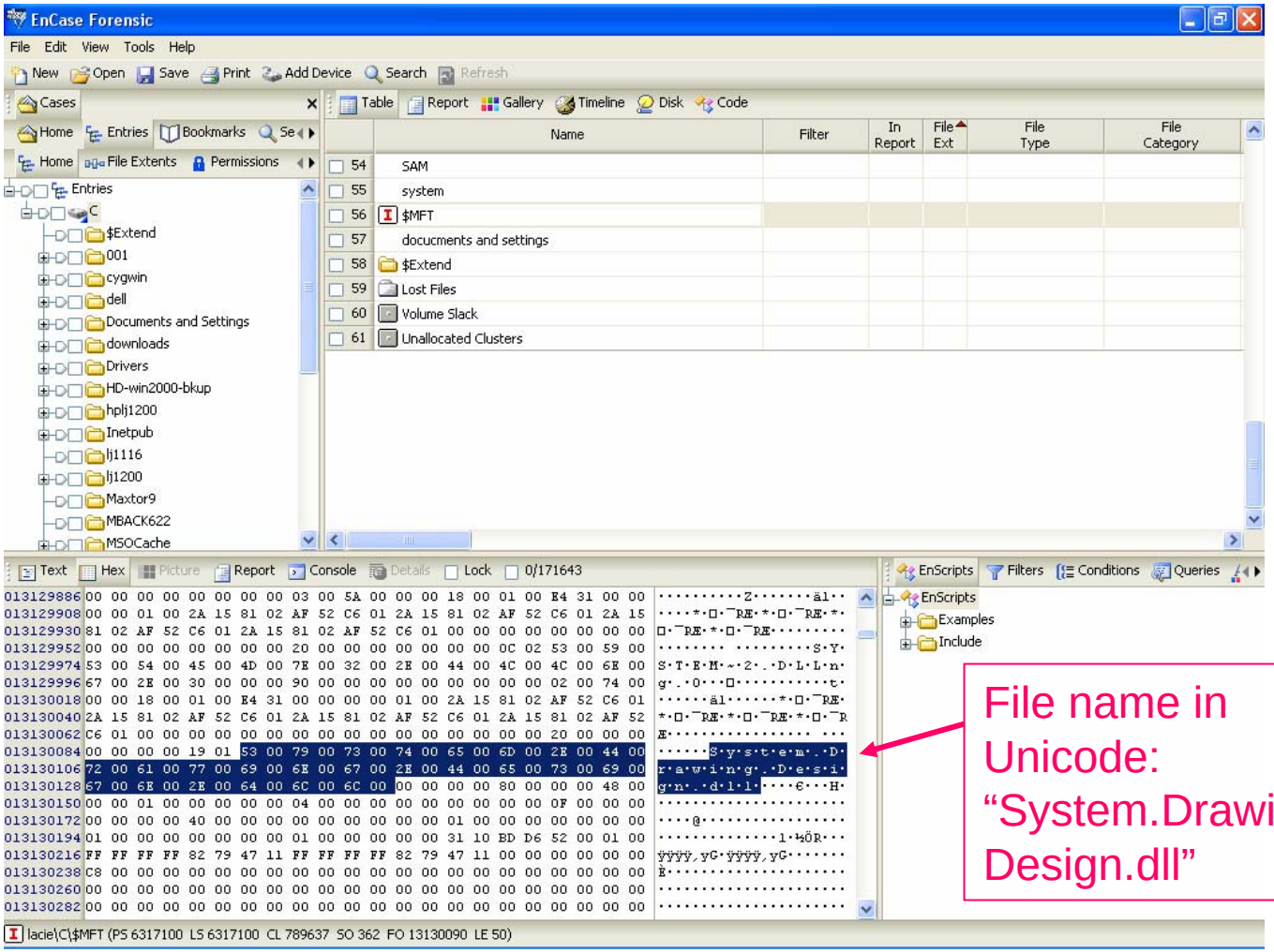
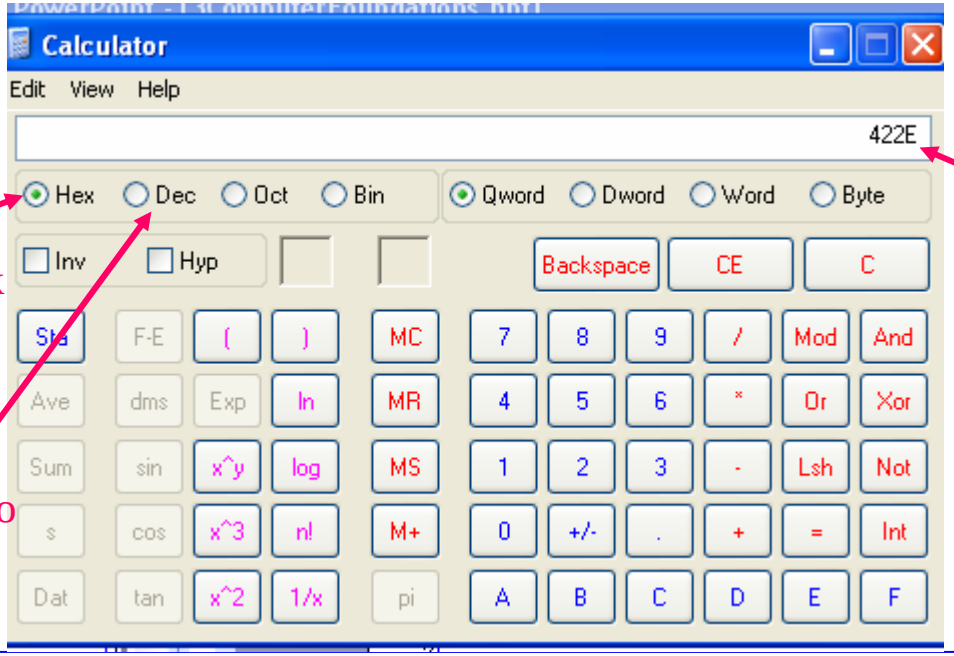
EnCase exam screenshot showing a file's name in MFT (master file table) of the NTFS file system:



Number system:

- decimal system: e.g., $123 = 1 * 10^2 + 2 * 10^1 + 3 * 10^0$
- binary system: e.g., $1101 = 1*2^3 + 1*2^2 + 0*2^1 + 1*2^0 = 13$ (in decimal)
- hexadecimal system: a base-16 system as an abbreviation for binary system by grouping 4 bits into a hex digit (0 – 9, A – F), e.g., $4C2E = 4*16^3 + 12*16^2 + 2*16^1 + 14*16^0 = 19502$ (in decimal, converted using Windows calculator pad)



Example. An IDE hard disk of size 60 GB (i.e., 60 billion bytes), which has a single partition of size 57207.9 MB (Megabytes). Why is the difference in size?
Answer: 1 KB = 1,024 bytes = 2^{10} bytes; 1 MB = 2^{20} bytes = 1048576 bytes. Thus, 57207.9 * 1048576 = 59986830950.4 ≈ 60 billion bytes

Representing negative numbers:
• Two common ways of representing negative numbers are sign-magnitude representation and two's-complement notation.
Note: Computers generally use two's-complement notation.
• Sign-magnitude sets the first bit to 1 if it is a negative number (the first bit, then, is not part of the quantity).
• Two's Complement inverts all of the bits and then adds one.

An example of binary arithmetic:

0000 1101 (13 in decimal)
1000 1101 (–13 in signed-magnitude notation)

1111 0010 (invert 0's and 1's for 13)
1111 0011 (2's complement representation of –13)
(note that –(–13) = 13, in both representations)

Subtraction works by adding the two's complement of a number
(and ignore carry out of the most significant place):

```
  25      0001 1001
–13   + 1111 0011
------   ----------
  12   10000 1100
         ignored
```

Organization of Multiple-Byte Non-Negative Numbers:

• Numbers larger than 255 require more than 1 byte to represent. Common sizes are 1-byte, 2-bytes (16-bit), 4-bytes (32-bit), and 8-bytes (64-bit).

• Consider the number 954 :

954 = 0000 0011 1000 1010 (using 16 bits) = 0x03BA (in hexadecimal), with the high (more significant) byte 03 and low byte BA,

- in Big Endian notation, 03 BA (high byte 03 first, then BA), as in Motorola processors, Sun Solaris, and IBM mainframe computers
- in Little Endian, BA 03 (low byte first in the lower address), as in Intel processors.