"Little Endian" and "Big Endian"

Depending on which computing system you use, you will have to consider the byte order in which multibyte numbers are stored, particularly when you are writing those numbers to a file. The two orders are called "Little Endian" and "Big Endian".

"Little Endian" means that the low-order byte of the number is stored in memory at the lowest address, and the high-order byte at the highest address. (The little end comes first.) For example, a 4 byte LongInt

```
Byte3 Byte2 Byte1 Byte0
```

will be arranged in memory as follows:

```
Base Address+0   Byte0
Base Address+1   Byte1
Base Address+2   Byte2
Base Address+3   Byte3
```

Intel processors (those used in PC's) use "Little Endian" byte order.

"Big Endian" means that the high-order byte of the number is stored in memory at the lowest address, and the low-order byte at the highest address. (The big end comes first.) Our LongInt, would then be stored as:

```
Base Address+0   Byte3
Base Address+1   Byte2
Base Address+2   Byte1
Base Address+3   Byte0
```

Motorola processors (those used in Mac's), Sun, SGI, IBM architectures use "Big Endian" byte order.