1.OutputStream的write（int b）方法为什么b是int型且每次读取一个字节，且将b的低8位转换为？

2.InputStream 的read（）和read(byte[])方法往往读取不到我们想要的字节数，如果要达到目的应该怎么做？

3. .InputStream 的read（）读取文件时，比如MP3往往会让文件损坏，而read(byte[])不会，这是为什么？

答:我估计和字节流读取中文乱码的原因一样，这样我们估计就需要重新编码。

4. .InputStream 的read（）和read(byte[])读取文件时具体过程是甚麽？

答：read（）每次读取一个字节并返回一个int（这个int返回值究竟是甚麽？如果是因为他要将读取的字节转返回出来他为什么不直接放回一个非-1的值不就行了吗？），后者需要注意一个问题就是，字节数组并不会被清空。

（常识补充：

使用哪些字符。也就是说哪些汉字，字母和符号会被收入标准中。所包含“字符”的集合就叫做“字符集”。

规定每个“字符”分别用一个字节还是多个字节存储，用哪些字节来存储，这个规定就叫做“编码”。

）

5. .InputStream 的read（）读取一个字节，为什么会返回一个int？

答：因为字节输入流可以操作任意类型的文件,比如图片音频等,这些文件底层都是以二进制形式的存储的,如果每次读取都返回byte,有可能在读到中间的时候遇到111111111

                那么这11111111是byte类型的-1,我们的程序是遇到-1就会停止不读了,后面的数据就读不到了,所以在读取的时候用int类型接收,如果11111111会在其前面补上

                24个0凑足4个字节,那么byte类型的-1就变成int类型的255了这样可以保证整个数据读完,而结束标记的-1就是int类型

FileInputStream.read()返回一个unsigned byte [0 - 255],而java里面没有这个类型，所以用int接收。byte的范围是[-128,127]，所以如果read()返回的数在[128,255]的范围内时，则表示负数，即

(byte)128=-128

(byte)129=-127

(byte)255=-1

所以如果read()返回的是byte的话，那就会有负数。而"返回-1意味着结束"，这个信息量用byte是无法表达的，所以必须用int。

（！！！！！！！！！！！注意：Java中没有无符号数的概念）。

{

byte与int的转换：byte(-1)=int(-1)

}

6.Java io流读写的数据会经过内存吗？

答：看是哪个流了。