**Base64**  
参考文章：https://blog.csdn.net/qq\_26093511/article/details/78836087  
**简介：**  
Bse64是一种以64个可打印字符对二进制数据进行编码的编码算法。base64在对数据进行编码时以三个8位字符型数据为一组，取这三个字符型数据的ASCII码，然后以6位为一组组成4个新的数据，这4个新的数据有6位，所以它的最大值为2^6=64。我们以4个6位数据的十进制数从base64表中得到最终编码后的字符。

**编码表：**

表1

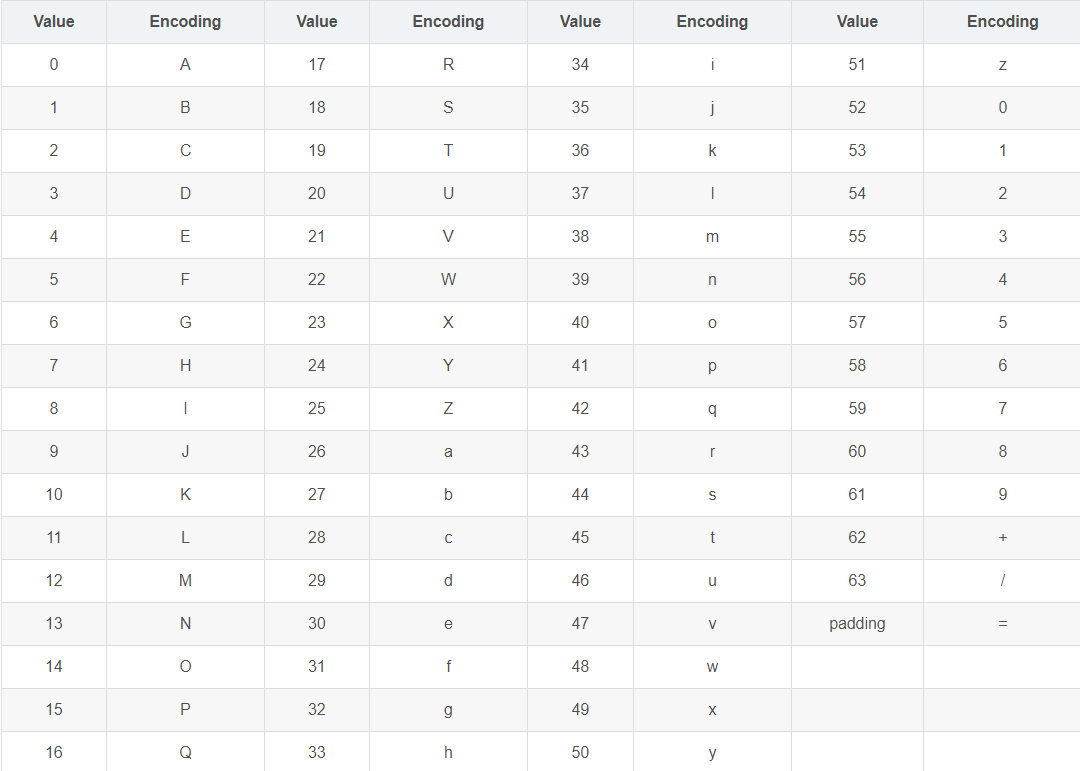
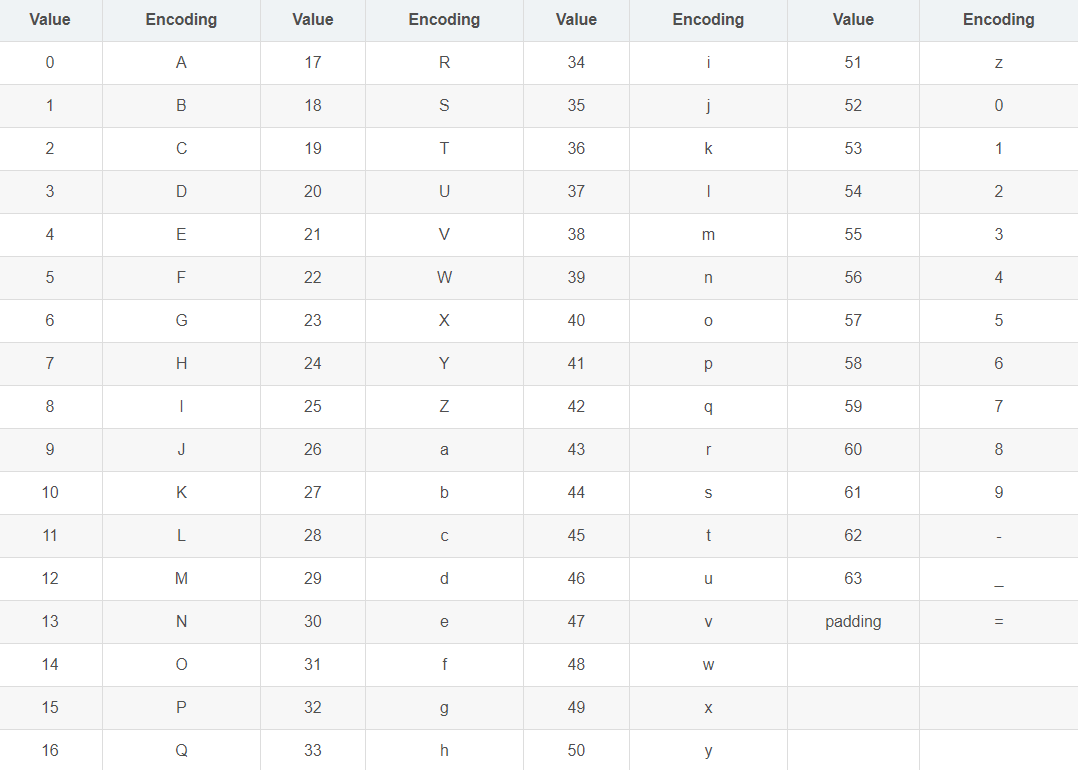


表2



由于base64编码是将编码前的3\*8位数据，分解成4个6位的数据，所以经过base64编码后的字符串长度是4的倍数。但往往我们进行编码的数据长度并不是3的倍数，这就造成了“编码”后的位数不为4的倍数，比如Brisk共5×8=40位，以6位为一组可以分为7组，这样“编码”后就有7个字符，但base64编码后的字符长度应该是4的倍数，显然这里就出问题了，那么怎么办呢？前面的不可以抛弃掉，所以就只有“追加”了，所以Brisk经过base64编码后的长度应该是8个字符，而第8个编码后的字符是'='，再比如对单个字符a进行base64编码，由于它的长度不是3的倍数，以3个字节为一组它只能分一组，再以6位为一位它只能分两组，所以经过“编码”后它的长度是2，但base64编码后的个数应该是4的倍数，所以它的长度应该是4，所以在后面补上两个‘=’,由于一个数求余3后有三个不同的结果，0、1、2，所以在对一个数据进行base64进行编码后它的长度为：  
（1）当进行编码的数据长度是3的倍数时，len=strlen(str\_in)/3\*4;  
（2）当进行编码的数据长度不是3的倍数时，len=([strlen(str\_in)/3]+1)\*4;  
  
 我们以Brisk这个例子来说明一下base64编码的过程。首先我们以3个字符为一组将Brisk进行分组，Brisk被分为两组：Bri和sk；然后我们取出这两个分组中每个字节的ASCII码，B:66 r:114 i:105 s:115 k:107。它们对应的二进制数为 B:01000010 r:01110010 i:01101001 s:01110011 k:01101011；  
  
  第一组，我们以6位为一组对每一个3字节分组进行再分组就变成了010000 100111 001001 101001。所对应的十进制数是16 39 9 41，对应base64表中的结果是 Q n J p；  
  第二组，011100 110110 101100(不够补0)，所以对应的十进制数是 28 54 44，对应base64表中的结果是 c 2 s，最终结果为QnJpc2s=（因为第二组“编码”后只有三个字节）。  
  
  解码的过程是一个逆过程，我们将经过编码后的字符按4个字符为一组，然后对照base64表得到相应的十进制数，再将其通过拆分和组合，组成3个8位数据，这个数据就是解码后的数据。

