输出：一个整数 x。

动作：将一个 8 位字节串 $M=M_0M_1\cdots M_{mlen-1}$ 转换为整数 x 方法如下：

将 M_i 看作[0～255]中的一个整数

$$x=\sum_{i=0}^{mlen-1}2^{8(mlen-1-i)}M_i$$

输出 x。

7 数据格式

7.1 密钥数据格式

SM2 算法私钥数据格式的 ASN.1 定义为：

```
SM2PrivateKey ::= INTEGER
```

SM2 算法公钥数据格式的 ASN.1 定义为：

```
SM2PublicKey ::= BIT STRING
```

SM2PublicKey 为 BIT STRING 类型，内容为 04 ‖ X ‖ Y，其中，X 和 Y 分别标识公钥的 x 分量和 y 分量，其长度各为 256 位。

7.2 加密数据格式

SM2 算法加密后的数据格式的 ASN.1 定义为：

```
SM2Cipher ::= SEQENCE{
XCoordinate        INTEGER,                  --x 分量
YCoordinate        INTEGER,                  --y 分量
HASH               OCTET STRING SIZE(32),    --杂凑值
CipherText         OCTET STRING              --密文
}
```

其中，HASH 为使用 SM3 算法对明文数据运算得到的杂凑值，其长度固定为 256 位。CipherText 是与明文等长的密文。

7.3 签名数据格式

SM2 算法签名数据格式的 ASN.1 定义为：

```
SM2Signature ::= {
R                  INTEGER,                  --签名值的第一部分
S                  INTEGER                   --签名值的第二部分
}
```

R 和 S 的长度各为 256 位。

7.4 密钥对保护数据格式

在 SM2 密钥对传递时，需要对 SM2 密钥对进行加密保护。具体的保护方法为：

a) 产生一个对称密钥。

b) 按对称密码算法标识指定的算法对 SM2 私钥进行加密，得到私钥的密文。若对称算法为分组算法，则其运算模式为 ECB。

c) 使用外部 SM2 公钥加密对称密钥得到对称密钥密文。

d) 将私钥密文、对称密钥密文封装到密钥对保护数据中。

SM2 密钥对的保护数据格式的 ASN.1 定义为：

```
SM2EnvelopedKey ::=SEQUENCE{
symAlgID                AlgorithmIdentifier,        --对称密码算法标识
symEncryptedKey         SM2Cipher,                  --对称密钥密文
Sm2PublicKey            SM2PublicKey,               --SM2 公钥
Sm2EncryptedPrivateKey  BIT STRING                  --SM2 私钥密文
}
```

8 预处理

8.1 预处理 1

预处理 1 是指使用签名方的用户身份标识和签名方公钥，通过运算得到 Z 值的过程。Z 值用于预处理 2，也用于 SM2 密钥协商协议。

输入：	ID	字节串	用户身份标识
	Q	SM2PublicKey	用户的公钥
输出：	Z	字节串	预处理 1 的输出

计算公式为：

$$Z = SM3(ENTL \| ID \| a \| b \| x_G \| y_G \| x_A \| y_A)$$

其中：

ENTL　为由 2 个字节表示的 ID 的比特长度；

ID　为用户身份标识；

a、b　为系统曲线参数；

x_G、y_G　为基点；

x_A、y_A　为用户的公钥。

详细的计算过程参见 GM/T 0003 和 GM/T 0004。

8.2 预处理 2

预处理 2 是指使用 Z 值和待签名消息，通过 SM3 运算得到杂凑值 H 的过程。杂凑值 H 用于 SM2 数字签名。

输入：	Z	字节串	预处理 2 的输入
	M	字节串	待签名消息
输出：	H	字节串	杂凑值

计算公式为：

$$H = SM3(Z \| M)$$

详细的计算过程见 GM/T 0003 和 GM/T 0004。

9 计算过程

9.1 生成密钥

SM2 密钥生成是指生成 SM2 算法的密钥对的过程，该密钥对包括私钥和与之对应的公钥。其中，私钥的长度为 256 位，公钥的长度为 512 位。

输入：　无

5.2 SM2 公钥

SM2 公钥是 SM2 曲线上的一个点，由横坐标和纵坐标两个分量来表示，记为(x,y)，简记为 Q，每个分量的长度为 256 位。

6 数据转换

在 SM2 算法的使用中将涉及 8 位字节串(Octet String)和位串(Bit String)之间的转换，主要包括以下四种形式。

6.1 位串到 8 位字节串的转换

位串长度若不是 8 的整数倍，需先在它的左边补 0，以保证它的长度为 8 的倍数，然后构造 8 位字节串，转换过程如下：

输入：一个长度为 blen 的位串 B。

输出：一个长度为 mlen 的字节串 M，其中 mlen 的取值为(blen+7)/8 的整数部分。

动作：将位串 $B=B_0B_1\cdots B_{blen-1}$ 转换到 8 位字节串 $M=M_0M_1\cdots M_{mlen-1}$ 采用如下方法：

从 $0\leqslant i\leqslant mlen-1$，设置：

$M_i=B_{blen-8-8(mlen-1-i)}B_{blen-7-8(mlen-1-i)}\cdots B_{blen-1-8(mlen-1-i)}$

对于 M_0，最左边 8-blen%8 位设置为 0，右边设置为 $B_0B_1\cdots B_{8-8mlen+blen-1}$。

输出 M。

6.2 8 位字节串到位串的转换

8 位字节串到位串转换过程如下：

输入：一个长度为 mlen 的 8 位字节串 M。

输出：一个长度为 blen=(8 * mlen)的位串 B。

动作：将 8 位字节串 $M=M_0M_1\cdots M_{mlen-1}$ 转换到位串 $B=B_0B_1\cdots B_{blen-1}$ 采用如下方法：

从 $0\leqslant i\leqslant mlen-1$，设置：$B_{8i}B_{8i+1}\cdots B_{8i+7}=M_i$

输出 B。

6.3 整数到 8 位字节串的转换

一个整数转换为 8 位字节串，基本方法是将其先使用二进制表达，然后把结果位串再转换为 8 位字节串。以下是转换流程：

输入：一个非负整数 x，期望的 8 位字节串长度 mlen。基本限制为：

$$2^{8(mlen)}>x$$

输出：一个长度为 mlen 的 8 位字节串 M。

动作：将基于 $2^8=256$ 的 x 值 $x=x_{mlen-1}2^{8(mlen-1)}+x_{mlen-2}2^{8(mlen-2)}+\cdots+x_1 2^8+x_0$ 转换为一个 8 位字节串 $M=M_0M_1\cdots M_{mlen-1}$ 采用如下方法：

从 $0\leqslant i\leqslant mlen-1$，设置：$M_i=x_{mlen-1-i}$

输出 M。

6.4 8 位字节串到整数的转换

可以简单地把 8 位字节串看成以 256 为基表示的整数，转换过程如下：

输入：一个长度 mlen 的 8 位字节串 M。