在C语言中，采用相对古老的随机数产生算法。在stdlib.h库中可以了解到，生成的随机数周期为0x7fff，也就是32767

C++11中引入了新的随机数引擎 mt19937。

Mersenne Twister算法译为马特赛特旋转演算法，是[伪随机数](https://baike.baidu.com/item/%E4%BC%AA%E9%9A%8F%E6%9C%BA%E6%95%B0" \t "_blank)发生器之一，其主要作用是生成伪随机数。此算法是Makoto Matsumoto （松本）和Takuji Nishimura （西村）于1997年开发的，基于有限二进制字段上的矩阵线性再生。可以快速产生高质量的伪随机数，修正了古老随机数产生算法的很多缺陷。

Mersenne Twister这个名字来自周期长度通常取Mersenne质数这样一个事实。常见的有两个变种Mersenne Twister MT19937和Mersenne Twister MT19937-64。

Mersenne Twister算法的原理：Mersenne Twister算法是利用[线性反馈移位寄存器](https://baike.baidu.com/item/%E7%BA%BF%E6%80%A7%E5%8F%8D%E9%A6%88%E7%A7%BB%E4%BD%8D%E5%AF%84%E5%AD%98%E5%99%A8/12731198" \t "_blank)(LFSR)产生随机数的，LFSR的反馈函数是寄存器中某些位的简单异或，这些位也称之为抽头序列。一个n位的LFSR能够在重复之前产生2^n-1位长的[伪随机序列](https://baike.baidu.com/item/%E4%BC%AA%E9%9A%8F%E6%9C%BA%E5%BA%8F%E5%88%97/4932210" \t "_blank)。只有具有一定抽头序列的LFSR才能通过所有2^n-1个内部状态，产生2^n - 1位长的伪随机序列，这个输出的序列就称之为m序列。为了使LFSR成为最大周期的LFSR，由抽头序列加上常数1形成的多项式必须是[本原多项式](https://baike.baidu.com/item/%E6%9C%AC%E5%8E%9F%E5%A4%9A%E9%A1%B9%E5%BC%8F/3246261" \t "_blank)。一个n阶本原多项式是不可约多项式，它能整除x^(2\*n-1)+1而不能整除x^d+1，其中d能整除2^n-1。例如(32,7,5,3,2,1,0)是指本原多项式x^32+x^7+x^5+x^3+x^2+x+1，把它转化为最大周期LFSR就是在LFSR的第32，7，5，2，1位抽头。利用上述两种方法产生周期为m的伪随机序列后，只需要将产生的伪随机序列除以序列的周期，就可以得到(0，1)上均匀分布的伪随机序列了。

Mersenne Twister有以下优点：随机性好，在计算机上容易实现，占用内存较少(mt19937的C程式码执行仅需624个字的工作区域)，与其它已使用的伪随机数发生器相比，产生随机数的速度快、周期长，可达到2^19937-1，且具有623维均匀分布的性质，对于一般的应用来说，足够大了，序列关联比较小，能通过很多随机性测试。

马特赛特旋转演算法产生一个伪随机数，一般为MtRand()。