

类似以下的错误信息：

```
Test detected that (-6.21,7.09) * ((2.52,-3.58) + (-4.51,3.91))
    != (-6.21,7.09) * (2.52,-3.58) + (-6.21,7.09) * (-4.51,3.91)
terminate called after throwing 'distributivity_violated'
```

接下来我们会深入到随机数发生过程的具体细节中。

4.2.2.3 引擎 (Engines)

库<random>包含两大类函数对象：发生器和分布器。前者可以生成一个无符号整数的序列（具体的类型在每个类中由 typedef 指定）。每个值被生成出来的概率大致相等。分布器类型则将值重新映射，以满足指定的分布所需要的概率。

除非有特殊需求，通常都可以使用 default_random_engine。记住，随机数序列明确依赖于它的种子，并且每个随机数发生引擎的对象在创建时都使用相同的种子进行初始化。因此新创建的相同类型的随机数发生引擎总是会返回同样的序列，例如：

```
void random_numbers()
{
    default_random_engine re;
    cout << "Random numbers: ";
    for (int i= 0; i < 4; i++)
        cout << re << (i < 3 ? ", " : "");
    cout << '\n';
}

int main ()
{
    random_numbers();
    random_numbers();
}
```

在作者的机器上会显示：

```
Random numbers: 16807, 282475249, 1622650073, 984943658
Random numbers: 16807, 282475249, 1622650073, 984943658
```

如果要求每次调用 random_numbers 都获得一个不同的序列，我们就需要将它的随机数发生引擎声明为 static：

```
void random_numbers()
{
    static default_random_engine re;
```