

随机数

```
1- rand 在<stdlib.h> 头文件中
2- 如果要每次产生不同的随机数
#include <stdlib.h>
#include <time.h>
srand((unsigned)time(NULL));
    for(int i = 0; i < 10;i++ )
        cout << rand() << '\t';

3- 产生[a,b] 之间的随机数
rand()%(b-a+1) + a;
```

计算机的随机数都是由伪随机数，即是由小M多项式序列生成的，其中产生每个小序列都有一个初始值，即随机种子。（注意：小M多项式序列的周期是**65535**，即每次利用一个随机种子生成的随机数的周期是**65535**，当你取得**65535**个随机数后它们又重复出现了。）

我们知道rand()函数可以用来产生随机数，但是这不是真正意义上的随机数，是一个伪随机数，是根据一个数（我们可以称它为种子，默认是设置为**1**）为基准以某个递推公式推算出来的一系列数，当这系列数很大的时候，就符合正态分布，从而相当于产生了随机数，但这不是真正的随机数，当计算机正常开机后，这个种子的值是定了的，除非你破坏了系统。

1- rand

功能:随机数发生器

用法:**int rand(void)**

所在头文件: **stdlib.h**

rand()的内部实现是用线性同余法做的，它不是真的随机数，因其周期特别长，故在一定的范围里可看成是随机的。

rand()返回一随机数值的范围在**0至RAND_MAX**[定义在 **stdlib.h**] 间。**RAND_MAX**的范围最少是在**32767**之间(int)。用unsigned int 双字节是65535，四字节是4294967295的整数范围。
0~RAND_MAX每个数字被选中的机率是相同的。

用户未设定随机数种子时，系统默认的随机数种子为**1**。

rand()产生的是伪随机数字，每次执行时是相同的;若要不同,用函数srand()初始化它。

```

#include<iostream>
#include <stdlib.h> /* RAND_MAX, int rand() */
using namespace std;

int main() {
    cout << RAND_MAX << endl;    // 2147483647

    // 产生10个随机数,每次运行产生的随机数是一样的
    for( int i=0; i<10;i++ )
        cout << rand() << " "; // 16807 282475249 1622650073 984943658
1144108930 470211272 101027544 1457850878 1458777923 2007237709
    return 0;
}

```

如何获取小数呢？例如：我们可以先获得10001以内的整数（0~10000），然后再用这个整数除以10000得到小数点后两位的小数。

```

#include<iostream>
using namespace std;
int main()
{
    for(int i=0;i<10;i++)
    {
        cout<<(rand()%10001)/10000.0<<" ";    // 0.6806 0.7005 0.7825
0.5174 0.4531 0.4256 0.7443 0.5108 0.206 0.7006
    }
}

```

2- srand

功能:初始化随机数发生器

用法: void srand(unsigned int seed)

所在头文件: stdlib.h

srand()用来设置rand()产生随机数时的随机数种子。参数seed必须是个整数，如果每次seed都设相同值，rand()所产生的随机数值每次就会一样。

rand()产生的随机数在每次运行的时候都是与上一次相同的。若要不同,用函数srand()初始化它。可以利用srand((unsigned int)(time(NULL)))的方法，产生不同的随机数种子，因为每一次运行程序的时间是不同的。

产生随机数的用法 1) 给srand()提供一个种子，它是一个unsigned int类型； 2) 调用rand()，它会根据提供给srand()的种子值返回一个随机数(在0到RAND_MAX之间)； 3) 根据需要多次调用rand()，从而不间断地得到新的随机数； 4) 无论什么时候，都可以给srand()提供一个新的种子，从而进一步“随机化”rand()的输出结果。

通常可以利用 `getpid()` 或 `time(0)` 的返回值来当做seed

如果你用time(0)的话，要加入头文件 `#include<ctime>`

time(0)或者time(NULL)返回的是系统的时间（从1970.1.1午夜算起），单位：秒

```
#include <iostream>
#include <ctime>
using namespace std;

int main()
{
    int a;
    a=time(0); //time(0)返回的是系统的时间（从1970.1.1午夜算起），单位：秒
    1521858244,1521858255
    cout<<a<<endl;
    return 0;
}
```

0~RAND_MAX之间的随机数程序

```
#include <iostream>
#include <stdlib.h>
#include <time.h>
using namespace std;
int main()
{
    /// 每次运行的结果是不一样的。
    srand((unsigned)time(NULL));
    for(int i = 0; i < 10;i++ )
        cout << rand() << '\t'; // 1335455916    1655985415    758804785
    1474125609    110275024    113941007    1598575172    85008187    655973654
    1915642727
    cout << endl;
    return 0;
}
```

3- 产生一定范围的随机数

产生一定范围随机数的通用表示公式 要取得[a,b)的随机整数，使用 $(\text{rand}() \% (b-a)) + a$; 要取得[a,b]的随机整数，使用 $(\text{rand}() \% (b-a+1)) + a$; 要取得(a,b)的随机整数，使用 $(\text{rand}() \% (b-a)) + a + 1$; 通用公式: $a + \text{rand}() \% n$; 其中的a是起始值，n是整数的范围。要取得a到b之间的随机整数，另一种表示： $a + (\text{int})b * \text{rand}() / (\text{RAND_MAX} + 1)$ 。要取得0~1之间的浮点数，可以使用 $\text{rand}() / \text{double}(\text{RAND_MAX})$ 。