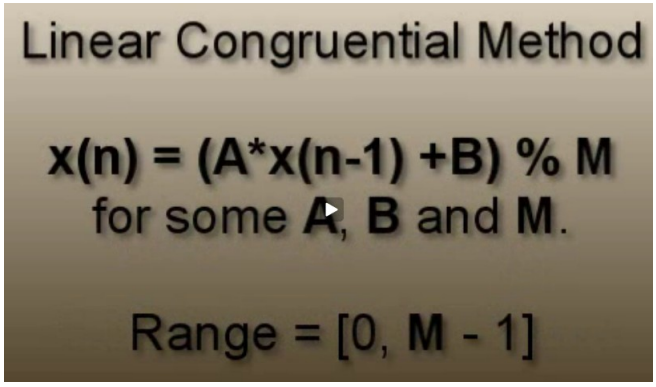
C++ Lesson 22 Using the rand() Function

◎參考網站 :

http://xoax.net/cpp/crs/console/lessons/Lesson22/

◎Video



在線性同餘法裡，選擇一個很大的整數 m，然後依下面的遞回方程式，創造出在 0 至 m-1 間之一連串整數群。

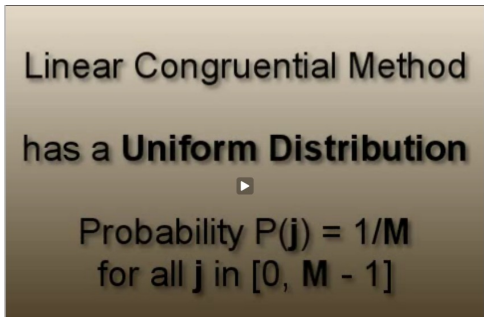
X(n-1) : 種子值 seed

A : 係數(固定乘數 constant multiplier)

B :增加量(increment)

M :模數(modulus) , M>0(除數)

◎Video



1. j 是 0~M-1 中的任一位數
2. P(j)是 j 被選上的可能性 , 它被選上的機率是 1/M

◎ rand()會返回一隨機數值,範圍在 0 至 RAND_MAX 間

```
int iCurr = 1;
const int RAND_MAX = 32767;
int rand(void) {
    iCurr = iCurr * 1134232345 + 24923;
    return((unsigned)(iCurr/65536) % 32768);
}
```

1. call rand()時, 當 seed 的 iCurr 自動設爲 1
2. iCurr 去乘 1 個很大的數 1134232345 , 再加 1 個大數 24923
3. 最後一行 , 任何正整數 mod 32768 的餘數 , 範圍是 0~32767
4. **懸疑: iCurr/65536**

◎ srand()

```
void srand(int iSeed) {
    iCurr = iSeed;
}
```

1. srand()用來設置 rand()的隨機數種子 seed。
2.Call rand()函數產生隨機數前，必須先利用 srand()設好隨機數種子（seed）, 如果未設亂數種子。
3.若沒有設置亂數種子 seed，則每次隨機數種子會自動設成 1，而導致 rand()每次所產生的亂數值都一樣。

```
//範例 1
//未 call srand() 範例
#include <iostream>
int main()
{
    using namespace std;
    for (int iIndex=0;iIndex<10;++iIndex){
        cout<<rand()<<" ";
    }
    system("pause");
    return 0;
}
```

```
//範例2
//call srand()範例
//call time()當 seed
#include <iostream>
#include <ctime>  //引用<ctime>標頭檔 , 因 time()函數定義在標頭檔 ctime 中
void main()
{
      using namespace std;
      time_t  t; //資料型別 int64 , 2^-64 ~ 2^64 之間的整數值
      time(&t); //相對於格林威治 1970 年 1 月 1 日 0 分 0 秒到現在的時間總秒數
      srand(t); // 只需 call srand()一次

      for(int iIndex=0;iIndex<10;++iIndex){
            cout<<rand()<<" "; //srand()會 call rand()傳入 t 當 seed ,而 rand()本身有 recurrsion 機制
      }
      cout<<endl;
      system("pause");
}
```

- 如果在程式中 call rand()時 , 後面加上模數運算子%以及模數 M , 則 rand() return 出來的亂數會再 mod M , 最後產生的結果亂數 , 會介於 0<=亂數<M
- 如例如 cout<<rand() %4<<" "; //所產生的亂數是 0~3

```
//範例 3
//產生 9 個 10~100(含 10、100)間的亂數
//參考書 :掌握 C++程式設計 7-30 頁
//公式:int r=m+rand()%(n-m+1); //m<=r<=n
#include <iostream>
void main(){
      using namespace std;
   for(int i=0;i<=8;i++){
    cout<<10+rand()%(100-10+1)<<; //最小數是 10+0 , 最大數是 10+90
   }
      system(“pause”);
}
```

```
//範例 4
//參考書 :掌握 C++程式設計 7-30 頁
//擲骰子遊戲
#include <iostream>
#include <ctime>  //引用<ctime>標頭檔
#include <conio.h>  //getch()引用<conio.h>標頭檔
using namespace std;
int main()
{
    srand((unsigned int)time(NULL)); //以系統時間當亂數種子
    int n;
    while(true) //無窮迴圈
    {
        cout << "請按任意鍵擲骰子  ";
        char ch=getch();  //按鍵
        if (ch == '\r')   //按 Enter 結束
        {
            cout << "擲骰子遊戲結束！\n";
            break;
        }
        else
        {
            n=1+rand()%(6-1+1);  //亂數 1~6
            cout << "點數爲：" << n << " 點\n";
         }
    }
    system("pause");
    return 0;
}
```

◎線性同餘法(LCG，Linear Congruential Method)介紹

http://tw.myblog.yahoo.com/jk3527101/article?mid=134&next=133&l=a

線性同餘法(LCG，Linear Congruential Method):
目前被廣泛使用的亂數產生方式，是發展歷史最久的。
最早是在1951年由Lehmer所提出。
在線性同餘法裡，選擇一個很大的整數m，然後依下面的遞回方程式，創造出在0至m-1間之一連串整數群。

recursion recursion

$$y_{i+1} = [ay_i + c] \bmod m$$

以下皆為整數:
y0 :種子值(seed)
a :固定乘數(constant multiplier)
c :增加量(increment)
m :模數(modulus)

y0是初始值，稱為種子值(seed)，其它的種子數yi是利用

$$y_{i+1} = [ay_i + c] \bmod m$$

依序產生出來，換言之就是自己產生的結果再拿來當種子數使用。
整個亂數列所產生的週期per(Ri)，將在per(Ri)<=m的條件下。
利用線性同餘法，當m=16、a=5、c=3、y0=7之週期:

recursion

i	yi	5yi	5yi+3	(5yi+3)/16	yi+1	Sol
0	7	35	38	38/16	6 y_1	0.375
1	6 y_1	30	33	33/16	1	0.063
2	1 y_2	5	8	8/16	8	0.5
3	8 y_3	40	43	43/16	11	0.688
.	.	.	.	.	.	.

線性同餘法範例:
參考下圖表格i=1~19的yi及sol，
由表格中可以看出y17=y1=6、y18=y2=1、y19=y3=8
由i=17、18、19與i=1、2、3相比較之下可以發現yi及sol有重複性。
因此可以定義i=1~16為一週期。

i	yi	sol	i	yi	sol	i	yi	sol	i	yi	sol
0	7		5	10	0.625	10	9	0.563	15	4	0.25
1	6	0.375	6	5	0.313	11	0	0	16	7	0.483
2	1	0.063	7	12	0.75	12	3	0.188	17	6	0.375
3	8	0.5	8	15	0.938	13	2	0.125	18	1	0.063
4	11	0.688	9	14	0.875	14	13	0.813	19	8	0.5

rand()與srand()運作原理與LCM比對推論

```
int iCurr = 1;
const int RAND_MAX = 32767;
int rand(void) {
    iCurr = iCurr * 1134232345 + 24923;
    return((unsigned)(iCurr/65536) % 32768);
}

void srand(int iSeed) {
    iCurr = iSeed;
}
```

y0是初始值，稱為種子值(seed)，其它的種子數yi是利用

$$y_{i+1} = [ay_i + c] \bmod m$$

利用線性同餘法，當m=16、a=5、c=3、y0=7之週期:

i 位數	yi	5yi	5yi+3	(5yi+3)/16	yi+1	Sol
0	7 y_0	35	38	38/16	6	0.375
1	6 y_1	30	33	33/16	1	0.063
2	1 y_2	5	8	8/16	8	0.5
3	8 y_3	40	43	43/16	11	0.688
.	.	.	.	.	.	.
.	.	.	.	.	.	.
.	.	.	.	.	.	.

◎參考網站 : 美麗 C 世界-亂數的使用
http://dhcp.tcgs.tc.edu.tw/c/p005.htm