

random

隨機



取隨機數

在C++中，取隨機數可以用函式庫<cstdlib>中的：

```
int rand();
```

回傳值：

介於 0 ~ RAND_MAX 間的偽隨機(pseudo-random)整數

取隨機數

在C++中，取隨機數可以用函式庫<cstdlib>中的：

```
int rand();
```

回傳值：

介於 0 ~ RAND_MAX 間的偽隨機(pseudo-random)整數

RAND_MAX保證至少是32767

偽隨機性 (Pseudorandomness)

偽隨機性是一個過程似乎是隨機的，但實際上並不是。例如偽亂數是使用一個**確定性的演算法**計算出來的似乎是隨機的數序，因此偽亂數實際上並不隨機。在計算偽亂數時假如使用的開始值不變的話，那麼偽亂數的數序也不變。

Source: Wikipedia - 偽隨機性

偽隨機性 (Pseudorandomness)

偽隨機性是一個過程似乎是隨機的，但實際上並不是。例如偽亂數是使用一個**確定性的演算法**計算出來的似乎是隨機的數序，因此偽亂數實際上並不隨機。在計算偽亂數時假如使用的開始值不變的話，那麼偽亂數的數序也不變。

簡單來說，由於偽隨機數是透過一個**確定性的演算法**來計算，因此「隨機數」實際上並不隨機

偽隨機性

取隨機數

在C++中，取隨機數可以用函式庫<cstdlib>中的：

```
int rand();
```

回傳值：

介於 0 ~ RAND_MAX 間的偽隨機(pseudo-random)整數

RAND_MAX保證至少是32767

取隨機數

範例：

```
1  #include <iostream>
2  #include <cstdlib>
3  using namespace std;
4
5  int main()
6  {
7      cout << rand();
8      return 0;
9  }
```

取隨機數

範例：

```
1  #include <iostream>
2  #include <cstdlib>
3  using namespace std;
4
5  int main()
6  {
7      cout << rand();
8      return 0;
9  }
```

在使用rand()之前，
必須先引入<cstdlib>函式庫

取隨機數

範例：

```
1  #include <iostream>
2  #include <cstdlib>
3  using namespace std;
4
5  int main()
6  {
7      cout << rand();
8      return 0;
9  }
```

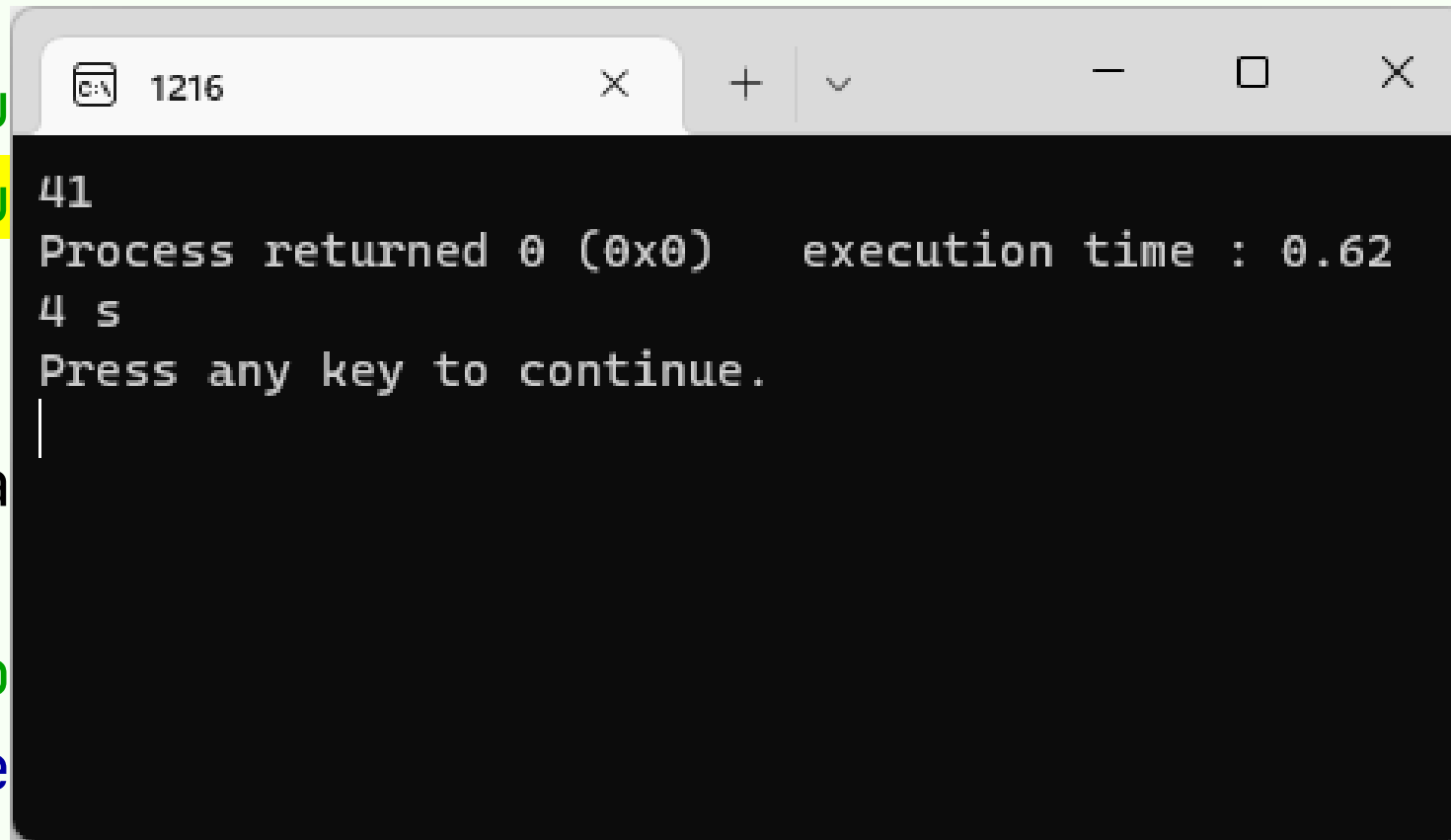
在使用rand()之前，
必須先引入<cstdlib>函式庫

利用rand()取得一個介於0 ~ RAND_MAX
的偽隨機(pseudo-random) 整數(int)

取隨機數

範例：

```
1 #include
2 #include
3 using
4
5 int ma
6 {
7     co
8     re
9 }
```



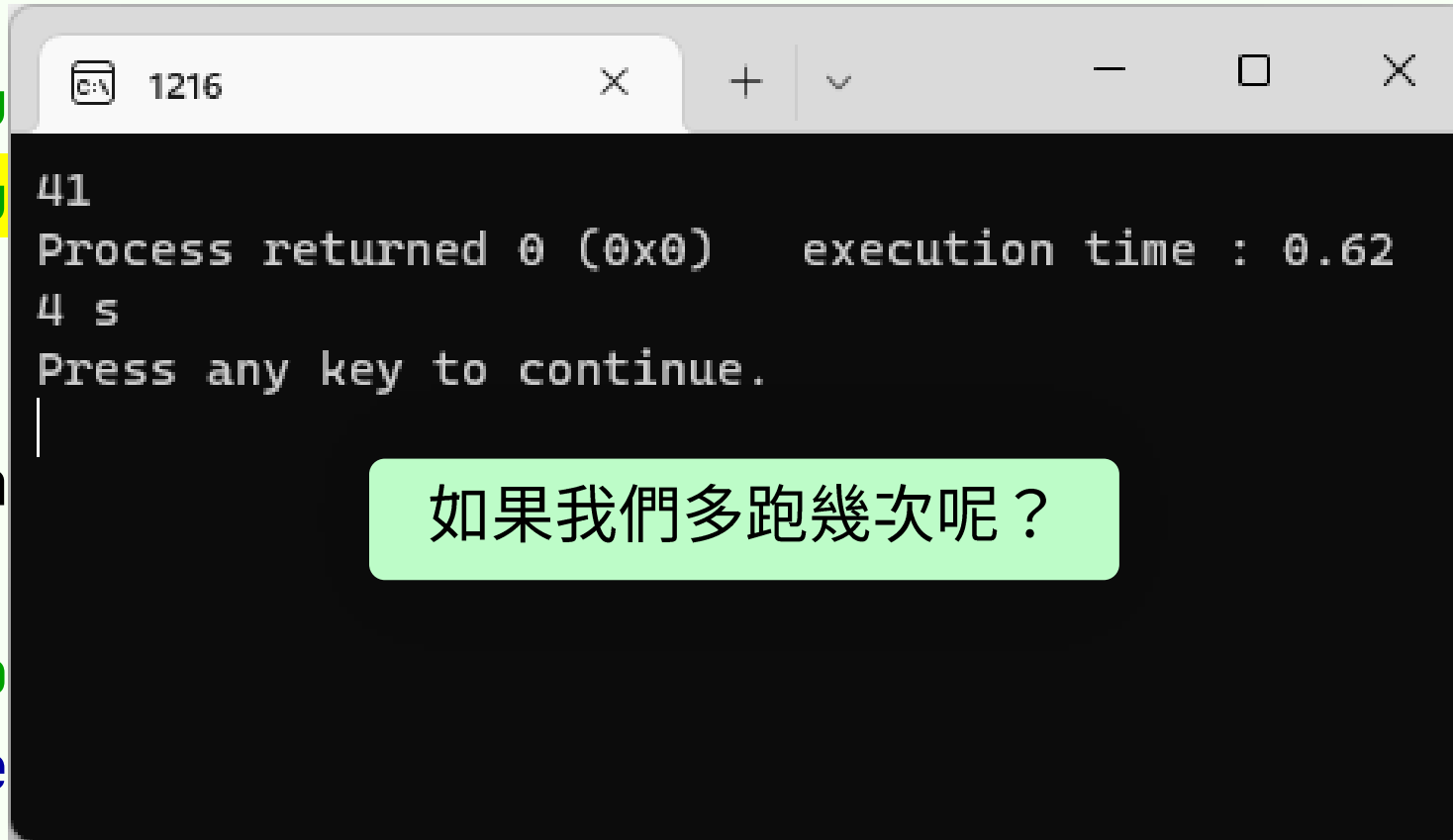
```
1216
41
Process returned 0 (0x0)   execution time : 0.62
4 s
Press any key to continue.
|
```

RAND_MAX
整數(int)

取隨機數

範例：

```
1 #include
2 #include
3 using
4
5 int ma
6 {
7     co
8     re
9 }
```



```
1216
41
Process returned 0 (0x0)   execution time : 0.62
4 s
Press any key to continue.
|
```

如果我們多跑幾次呢？

RAND_MAX
整數(*int*)

取隨機數

每次的結果都一樣！

```
41  
Process returned 0 (0x0)  
4 s  
Press any key to continue.
```

```
41  
Process returned 0 (0x0)  
4 s
```

```
41  
Process returned 0 (0x0)  
4 s  
Press any key to continue.
```

```
41  
Process returned 0 (0x0)  
4 s  
Press any key to continue.
```

```
41  
Process returned 0 (0x0)  
4 s  
Press any key to continue.
```

```
41  
Process returned 0 (0x0)  
4 s  
Press any key to continue.
```

取隨機數

需要重新設定一個亂數種子。可以利用srand函式：

```
void srand(unsigned seed);
```

取隨機數

範例：

```
1  #include <iostream>
2  #include <cstdlib>
3  #include <ctime>
4  using namespace std;
5
6  int main()
7  {
8      srand(time(0));
9      cout << rand();
10     return 0;
11 }
```

取隨機數

範例：

```
1  #include <iostream>
2  #include <cstdlib>
3  #include <ctime>
4  using namespace std;
5
6  int main()
7  {
8      srand(time(0));
9      cout << rand();
10     return 0;
11 }
```

使用srand函式來重新設定亂數種子

通常會用現在的時間來當亂數種子

取隨機數

範例：

```
1  #include <iostream>
2  #include <cstdlib>
3  #include <ctime>
4  using namespace std;
5
6  int main()
7  {
8      srand(time(0));
9      cout << rand();
10     return 0;
11 }
```

UNIX時間：
從UTC1970年1月1日0時0分0秒起至現在的總秒數，不考慮閏秒

使用time(0)來取得現在的UNIX時間

取隨機數

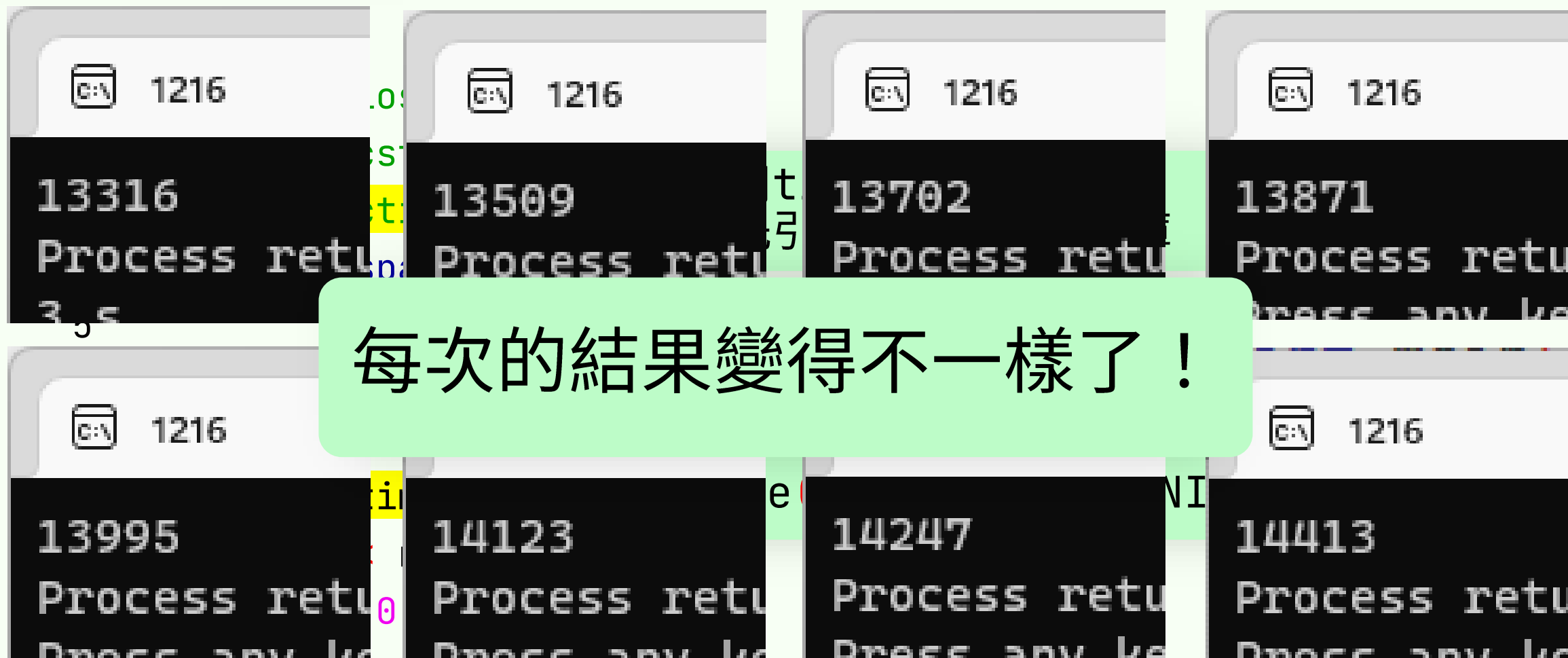
範例：

```
1  #include <iostream>
2  #include <cstdlib>
3  #include <ctime>
4  using namespace std;
5
6  int main()
7  {
8      srand(time(0));
9      cout << rand();
10     return 0;
11 }
```

在使用time(0)之前，
必須先引入<ctime>函式庫

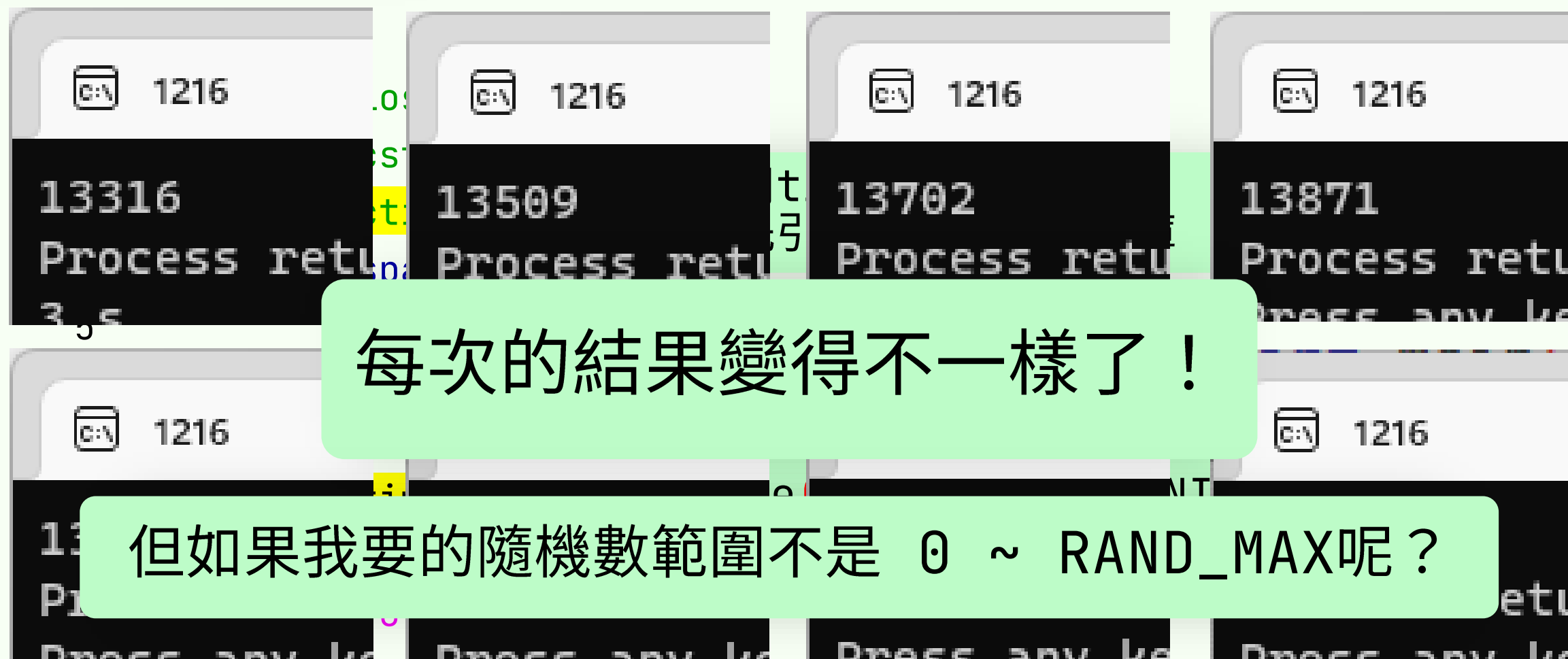
使用time(0)來取得現在的UNIX時間

取隨機數



每次的結果變得不一樣了！

取隨機數



每次的結果變得不一樣了！

但如果我要的隨機數範圍不是 0 ~ RAND_MAX呢？

取特定範圍整數隨機數

可以使用取餘運算子 **%** 來限定rand()的範圍，例如：

rand() % 10 範圍：0 ~ 9

rand() % 35 範圍：0 ~ 34

rand() % N 範圍：0 ~ N-1

取特定範圍整數隨機數

可以使用取餘運算子 `%` 來限定`rand()`的範圍，例如：

`rand() % 10 + 1` 範圍：(0 + 1) ~ (9 + 1)

`rand() % 35` 範圍：0 ~ 34

`rand() % N` 範圍：0 ~ N-1

取特定範圍整數隨機數

可以使用取餘運算子 `%` 來限定`rand()`的範圍，例如：

`rand() % 10 + 1` 範圍：1 ~ 10

`rand() % 35` 範圍：0 ~ 34

`rand() % N` 範圍：0 ~ N-1

取特定範圍整數隨機數

可以使用取餘運算子 `%` 來限定`rand()`的範圍，例如：

`rand() % 10 + 1` 範圍：1 ~ 10

`rand() % 35 + 15` 範圍：(0 + 15) ~ (34 + 15)

`rand() % N` 範圍：0 ~ N-1

取特定範圍整數隨機數

可以使用取餘運算子 `%` 來限定`rand()`的範圍，例如：

`rand() % 10 + 1` 範圍：1 ~ 10

`rand() % 35 + 15` 範圍：15 ~ 49

`rand() % N` 範圍：0 ~ N-1

取特定範圍整數隨機數

可以使用取餘運算子 `%` 來限定`rand()`的範圍，例如：

`rand() % 10 + 1`

範圍：1 ~ 10

`rand() % 35 + 15`

範圍：15 ~ 49

`rand() % N + k`

範圍： $(0 + k) \sim (N - 1 + k)$

取特定範圍整數隨機數

可以使用取餘運算子 `%` 來限定`rand()`的範圍，例如：

`rand() % 10 + 1`

範圍：1 ~ 10

`rand() % 35 + 15`

範圍：15 ~ 49

`rand() % N + k`

範圍： $k \sim (N - 1 + k)$

取特定範圍整數隨機數

可以使用取餘運算子 `%` 來限定`rand()`的範圍，例如：

`rand() % N + k`

範圍： $k \sim (N - 1 + k)$

假設 $N' = (N - 1 + k)$ ，求N

$$N - 1 + k = N'$$

$$N - 1 = N' - k$$

$$N = N' - k + 1$$

取特定範圍整數隨機數

可以使用取餘運算子 $\%$ 來限定rand()的範圍，例如：

$\text{rand}() \% N + k$

範圍： $k \sim (N - 1 + k)$

假設 $N' = (N - 1 + k)$ ，求N

$$N - 1 + k = N'$$

$$N - 1 = N' - k$$

$$N = N' - k + 1$$

將 $N = N' - k + 1$ 代回原式

取特定範圍整數隨機數

將 $N = N' - k + 1$ 代回原式 來限定rand()的範圍，例如：

$\text{rand}() \% N + k$

範圍： $k \sim (N - 1 + k)$

取特定範圍整數隨機數

將 $N = N' - k + 1$ 代回原式 來限定rand()的範圍，例如：

$$\text{rand}() \% (N' - k + 1) + k$$

$$\text{範圍：} k \sim ((N' - k + 1) - 1 + k)$$

取特定範圍整數隨機數

將 $N = N' - k + 1$ 代回原式 來限定rand()的範圍，例如：

$$\text{rand}() \% (N' - k + 1) + k$$

$$\text{範圍：} k \sim (N' - k + 1 - 1 + k)$$

取特定範圍整數隨機數

將 $N = N' - k + 1$ 代回原式 來限定rand()的範圍，例如：

$$\text{rand}() \% (N' - k + 1) + k$$

$$\text{範圍：} k \sim (N' - k + 1 - 1 + k)$$

取特定範圍整數隨機數

將 $N = N' - k + 1$ 代回原式 來限定rand()的範圍，例如：

$$\text{rand}() \% (N' - k + 1) + k$$

範圍： $k \sim (N' - k + k)$

取特定範圍整數隨機數

將 $N = N' - k + 1$ 代回原式 來限定rand()的範圍，例如：

$$\text{rand}() \% (N' - k + 1) + k$$

範圍： $k \sim (N' - \cancel{k} + k)$

取特定範圍整數隨機數

將 $N = N' - k + 1$ 代回原式 來限定rand()的範圍，例如：

$$\text{rand}() \% (N' - k + 1) + k$$

範圍： $k \sim N'$

取特定範圍整數隨機數

可以使用取餘運算子 `%` 來限定`rand()`的範圍，例如：

`rand() % (N' - k + 1) + k`

範圍： $k \sim N'$

故如欲求範圍 $k \sim N'$ 的整數隨機數，

可以使用`rand() % (N' - k + 1) + k`

取特定範圍整數隨機數

可以使用取餘運算子 `%` 來限定`rand()`的範圍，例如：

`rand() % (Max - Min + 1) + Min`

範圍：`Min ~ Max`

故如欲求範圍 `Min ~ Max` 的整數隨機數，

可以使用`rand() % (Max - Min + 1) + Min`

取特定範圍整數隨機數

假設要產生的隨機數 x 的範圍要介於Min和Max之間
($Min \leq x \leq Max$)

則可以這樣寫：

```
int x = Min + rand() % (Max - Min + 1);
```

取特定範圍整數隨機數

範例：欲取1~10之間的隨機整數x
($1 \leq x \leq 10$)

則可以這樣寫：

```
int x = 1 + rand() % (10 - 1 + 1);
```

取特定範圍整數隨機數

```
1  #include <iostream>
2  #include <cstdlib>
3  #include <ctime>
4  using namespace std;
5
6  int main()
7  {
8      int Min = 1, Max = 10, x;
9
10     srand(time(0));
11     x = Min + rand() % (Max - Min + 1);
12
13     cout << x;
14     return 0;
15 }
```


取特定範圍整數隨機數

```
1  #include <iostream>
2  #include <cstdlib>
3  #include <ctime>
4  using namespace std;
5
6  int main()
7  {
8      int Min = 1, Max = 10, x;
9
10     srand(time(0));
11     x = Min + rand() % (Max - Min + 1);
12
13     cout << x;
14     return 0;
15 }
```

假設欲取1~10之間的隨機整數值

取特定範圍整數隨機數

```
1  #include <iostream>
2  #include <cstdlib>
3  #include <ctime>
4  using namespace std;
5
6  int main()
7  {
8      int Min = 1, Max = 10, x;
```

使用現在的UNIX時間作為亂數種子

```
    srand(time(0));
    x = Min + rand() % (Max - Min + 1);

    cout << x;
    return 0;
}
```

假設欲取1~10之間的隨機整數值

取特定範圍整數隨機數

```
1  #include <iostream>
2  #include <cstdlib>
3  #include <ctime>
4  using namespace std;
5
6  int main()
7  {
8      int Min = 1, Max = 10, x;
9
10
11
12
13
14
15 }
```

假設欲取1~10之間的隨機整數值

使用現在的UNIX時間作為亂數種子

```
srand(time(0));
```

```
x = Min + rand() % (Max - Min + 1);
```

```
cout << x;
```

```
return 0;
```

取Min~Max之間的
隨機整數值

取特定範圍整數隨機數

使用現在的UNIX時間作為亂數種子

```
1 #include <iostream>
```

9

```
2
```



1216



```
3
```

```
4
```

```
1
```

```
5
```

```
Process returned 0 (0x0)    execution time : 0.047 s
```

```
6
```

```
Press any key to continue.
```

```
7
```

```
8
```

```
+ 1);
```

之間的

取特定範圍整數隨機數

搭配自訂函式，可以這樣寫：

取特定範圍整數隨機數

搭配自訂函式，可以這樣寫：

```
int randint(int Min, int Max){  
    if (Min > Max){  
        int tmp = Max;  
        Max = Min;  
        Min = tmp;  
    }  
    return Min + rand() % (Max - Min + 1);  
}
```

取特定範圍整數隨機數

搭配自訂函式，可以 傳入欲取的範圍

```
int randint(int Min, int Max){  
    if (Min > Max){  
        int tmp = Max;  
        Max = Min;  
        Min = tmp;  
    }  
    return Min + rand() % (Max - Min + 1);  
}
```

取特定範圍整數隨機數

搭配自訂函式，可以這樣寫：

```
int randint(int Min, int Max){  
    if (Min > Max){  
        int tmp = Max;  
        Max = Min;  
        Min = tmp;  
    }  
    return Min + rand() % (Max - Min + 1);  
}
```

如果Min > Max

取特定範圍整數隨機數

搭配自訂函式，可以這樣寫：

```
int randint(int Min, int Max){  
    if (Min > Max){  
        int tmp = Max;  
        Max = Min;  
        Min = tmp;  
    }  
    return Min + rand() % (Max - Min + 1);  
}
```

則 Min 和 Max 交換

取特定範圍整數隨機數

搭配自訂函式，可以這樣寫：

```
int randint(int Min, int Max){  
    if (Min > Max){  
        int tmp = Max;  
        Max = Min;  
        Min = tmp;  
    }  
    return Min + rand() % (Max - Min + 1);  
}
```

取範圍內隨機整數值

取特定範圍整數隨機數

搭配自訂函式，可以這樣寫：

```
int randint(int Min, int Max){  
    if (Min > Max){  
        int tmp = Max;  
        Max = Min;  
        Min = tmp;  
    }  
    return Min + rand() % (Max - Min + 1);  
}
```

回傳範圍內隨機整數值

取特定範圍整數隨機數

搭配自訂函式，可以這樣寫：

```
int randint(int Min, int Max){
```

如此一來，當宣告完函式randint後，
只要呼叫randint(int Min, int Max)
就可以取特定範圍整數隨機數

```
    return Min + rand() % (Max - Min + 1);
```

```
}
```

取特定範圍整數隨機數

```
1  #include <iostream>
2  #include <cstdlib>
3  #include <ctime>
4  using namespace std;
5
6  int randint(int Min, int Max){
7      if (Min > Max){
8          int tmp = Max;
9          Max = Min;
10         Min = tmp;
11     }
12     return Min + rand() % (Max - Min + 1);
13 }
14
15 int main()
16 {
17     srand(time(0));
18     for (int i = 0; i < 10; i++){
19         cout << randint(3, 1) << endl;
20     }
21     return 0;
22 }
```

```
2 #include <stdlib.h>
```

取特定範圍整數隨機數

```
5  
6  int randint(int Min, int Max){  
7      if (Min > Max){  
8          int tmp = Max;  
9          Max = Min;  
10         Min = tmp;  
11     }  
12     return Min + rand() % (Max - Min + 1);  
13 }  
14
```

```
2 #include <stdlib.h>
```

取特定範圍整數隨機數

5

```
6 int randint(int Min, int Max){
```

```
7     if (Min > Max){
```

```
8         int tmp = Max;
```

```
9         Max = Min;
```

```
10        Min = tmp;
```

```
11    }
```

```
12    return Min + rand() % (Max - Min + 1);
```

```
13 }
```

14

宣告randint

取特定範圍整數隨機數

`% (Max - Min + 1);`

14

15 `int main()`

16 `{`

使用現在的UNIX時間作為亂數種子

17 `srand(time(0));`

18 `for (int i = 0; i < 10; i++){`

19 `cout << randint(3, 1) << endl;`

20 `}`

21 `return 0;`

22 `}`

取特定範圍整數隨機數

6 (Max - Min + 1);

```
14
15  int main()
16  {
17      srand(time(0));
18      for (int i = 0; i < 10; i++){
19          cout << randint(3, 1) << endl;
20      }
21      return 0;
22  }
```

重複循環10次

取特定範圍整數隨機數

`(Max - Min + 1);`

```
14
15  int main()
16  {
17      srand(time(0));
18      for (int i = 0; i < 10; i++){
19          cout << randint(3, 1) << endl;
20      }
21      return 0;
22  }
```

呼叫randint(3, 1)

取特定範圍整數隨機數

`% (Max - Min + 1);`

```
14
15  int main()
16  {
17      srand(time(0));
18      for (int i = 0; i < 10; i++){
19          cout << randint(3, 1) << endl;
20      }
21      return 0;
22  }
```

輸出

取特定範圍整數隨機數

$(\text{Max} - \text{Min} + 1);$

14

15 `int`

16 `{`

17

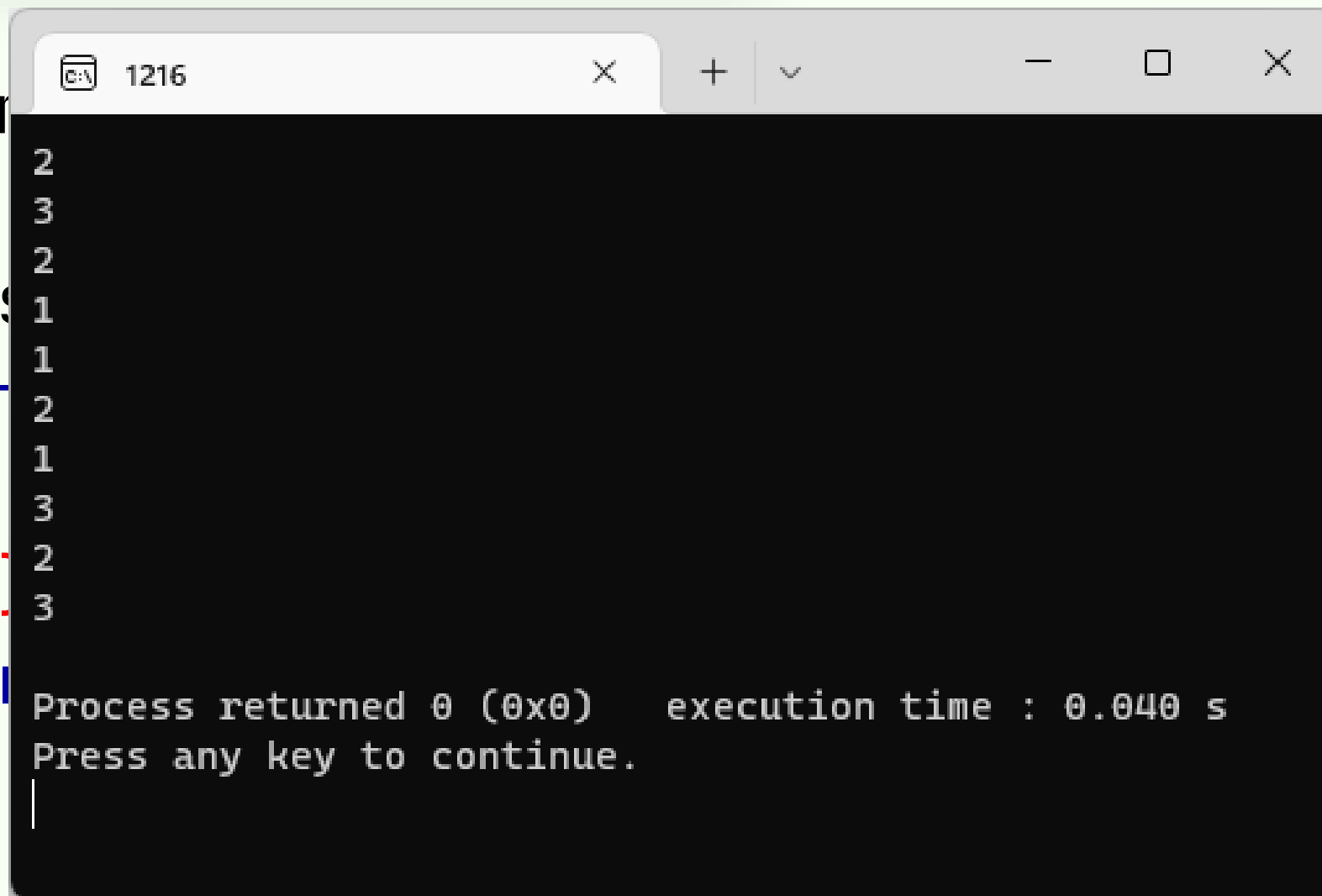
18

19

20

21

22 `}`



```
2
3
2
1
1
2
1
3
2
3

Process returned 0 (0x0)   execution time : 0.040 s
Press any key to continue.
|
```

110學年度 上機考試題

亂數判斷奇偶，並求其機率（建議使用陣列）

題目敘述：

請輸入一不大於10000的正整數N，求取50個範圍在 $1 \sim N$ 的亂數中，分別列出那些數值為奇數，那些數值為偶數，並算出其機率值。（參考所附執行檔）

例如 輸入:6666

奇數值有：

```
====> 2281, 2929, 2833, 3135, 17, 3089, 5063, 4599, 231, 2583,
====> 961, 2669, 921, 1151, 3319, 2853, 3629, 5383, 5421, 1569,
====> 5749, 2711, 1675, 79, 3379, 385, 6325,
====> 共有27個,佔機率為0.54
```

偶數值有：

```
====> 5678, 4296, 4662, 736, 2178, 5354, 870, 5238, 4802, 5954,
====> 746, 2466, 2642, 1608, 164, 6430, 4910, 4036, 1004, 1398,
====> 702, 3568, 5510,
====> 共有23個,佔機率為0.46
```