

第二章 終極密碼

知道機率和迴圈是甚麼概念後，就可以製作非常多的遊戲，只要想得道通常都做得出來，基本遊戲流程如下：

1. 基本概念



簡單來說，一開始是參數設定，要設定的內容跟遊戲有關；則設定好後會到玩家回合，玩家輸入行動，由電腦(手機)計算之後再回去等待玩家

<div><div>Coding C++RUNMENU</div><div>Auto saved at 02:19:51</div><pre>1 #include<bits/stdc++.h> 2 using namespace std; 3 int main(){ 4 srand(time(0)); 5 int things=1000; 6 for(;;){ 7 cout<<"想要付出多少 (剩下"<<things<<");"; 8 int x; 9 cin>>x; 10 things=things-x; 11 int number=rand()%100; 12 int percent=40; 13 if(number<percent){ 14 cout<<"得到 "<<x*0.5<<endl; 15 things=things+x*1.5; 16 } 17 else cout<<"沒中"<<endl; 18 if(things<=0){ 19 cout<<"東西沒了"<<endl; 20 break; 21 } 22 } 23 }</pre></div>	<p>以左邊的抽獎程式為例：</p> <p>第 4 ~ 5 行 → 參數設定</p> <p>第 7 ~ 9 行 → 玩家回合</p> <p>第10以後 → 電腦計算</p>
---	--

接下來呢，將會基於這種架構，製作新的遊戲【終極密碼】

2. 終極密碼

第一步當然先進行參數設定，要玩終機密碼一開始需要3個數字，分別是：答案(num)、下限(low)、上限(high)，就可以組合出這遊戲的第一句話「從 ?? 到 ?? 中猜個數字」程式如下：

```
Coding C++
Auto saved at 18:50:16
RUN MENU

1 #include<bits/stdc++.h>
2 using namespace std;
3 int main() {
4     srand(time(0));
5     int number=rand()%1000;
6     int low=0,high=1000;
7     for(;;){
8     }
9 }
```

第 5 行→ `number=rand()%1000` 代表從 0~999 中隨機取數值

第 6 行→ `int low=0,high=1000` 代表設定兩個數值

low 表示範圍下限，此時的下限是 0

high 表示範圍上限，此時的上限是 1000

第 7 & 8 行→ `for(;;){}`，在這迴圈裡面要放的就是 玩家回合 和 電腦計算

第一步的參數設定寫好之後進到第二步玩家回合，給出一個範圍讓玩家輸入數字開始猜，程式如下：

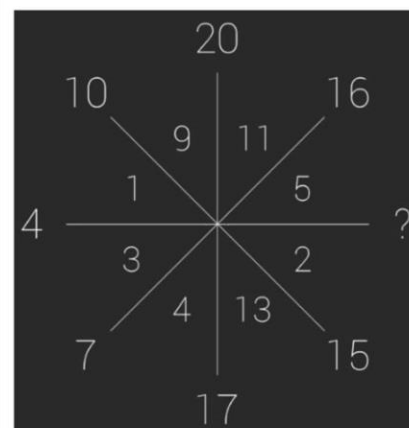
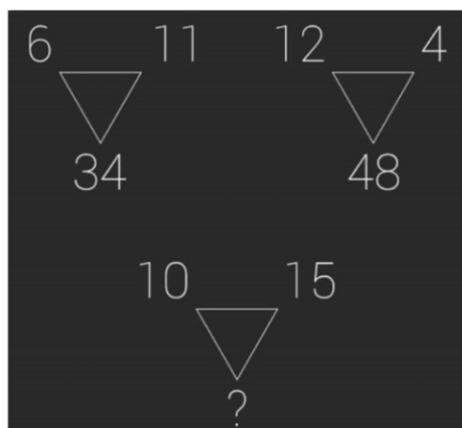
```
3 int main(){
4     srand(time(0));
5     int num=rand()%1000;
6     int low=0 high=1000;
7     for(;;){
8         cout<<low<<" 到 "<<high<<"中選一個數字:";
9         int x;cin>>x;
10    }
11 }
```

第 8 行→ 輸出的文字是：0 到 1000 中選一個數字，因為low=0 high=1000
第 9 行→ 玩家輸入一個數字進去當成 x 的數值

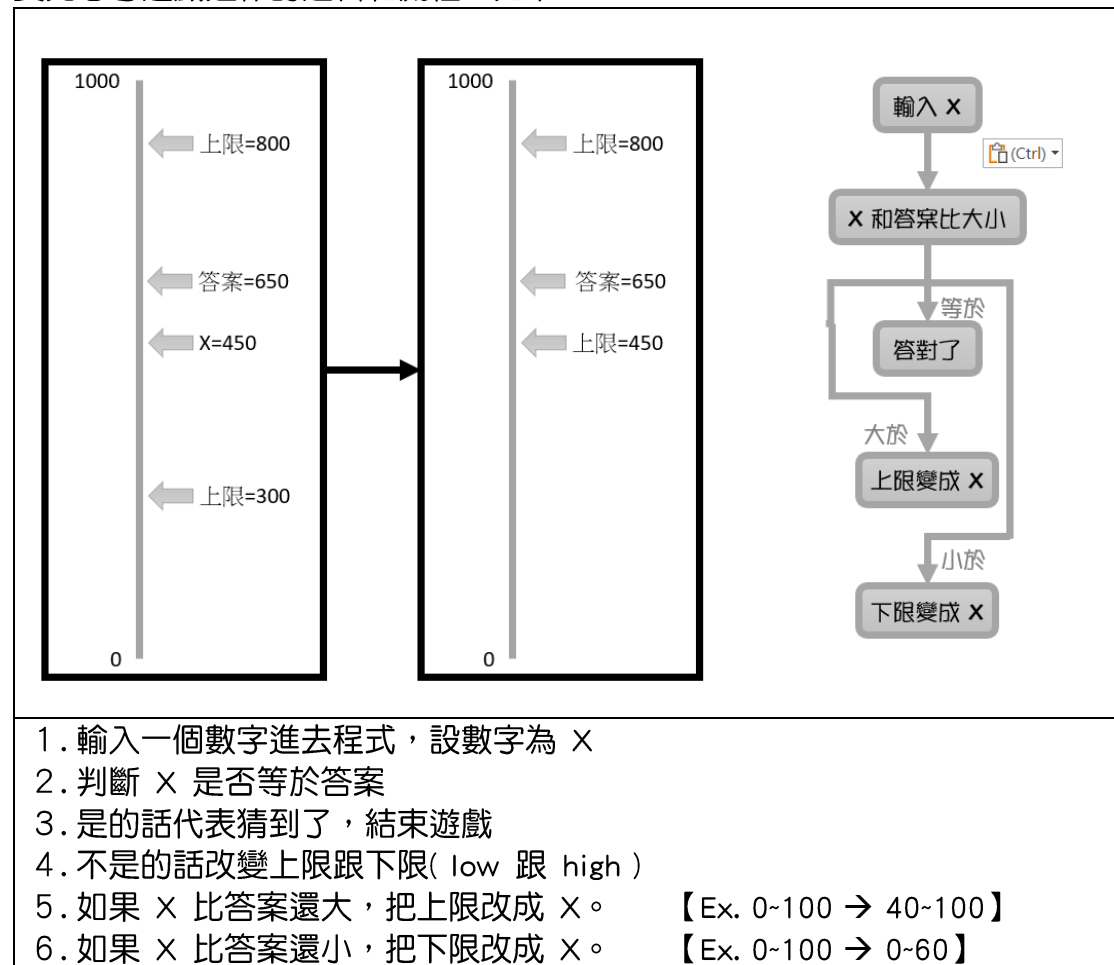
35	1	23	8	6	22	3	26
2	3	18	1	1	8	17	25
74	26	11	63	26	34		

[illegible]

3	20	11	7	1	3	29	18
3	14	5	16	19	49	11	



完成參數設定和玩家回合後，就可以接著處理電腦計算的部分，這一步都要先思考遊戲運作的邏輯和流程，如下：



下一步就是把流程寫成程式碼，解說再下一頁

```
7   for(;;){
8       cout<<low<<" 到 "<<high<<"中選一個數字:";
9       int x;cin>>x;
10      if(x==number){
11          cout<<"答對了"<<endl;
12          break;
13      }
14      if(x>number){
15          high=x;
16      }
17      if(x<number){
18          low=x;
19      }
20  }
21 }
```

第10行→ `if(x==number)` 表示 x 等於答案時，執行10~13中大括號的部分
第12行→ `break` 用來答對後會結束遊戲
第14行→ `if(x>number)` 表示 x 大於答案時，執行第15行
第15行→ `high=x` 代表「把上限改成 x 」
第16行→ `if(x<number)` 表示 x 小於答案時，執行第17行
第17行→ `low=x` 代表「把下限改成 x 」

那 x 跟答案的關係一定是大於、等於、小於的齊中一種，所以第10、14、16行都只會有一行是成立的，執行出來的成果如右：

```
0 到 1000中選一個數字：500
500 到 1000中選一個數字：750
500 到 750中選一個數字：625
500 到 625中選一個數字：575
500 到 575中選一個數字：544
500 到 544中選一個數字：522
522 到 544中選一個數字：533
522 到 533中選一個數字：526
526 到 533中選一個數字：530
526 到 530中選一個數字：528
答對了
```

```
0 到 1000中選一個數字：500
500 到 1000中選一個數字：750
750 到 1000中選一個數字：875
875 到 1000中選一個數字：922
875 到 922中選一個數字：900
900 到 922中選一個數字：911
900 到 911中選一個數字：906
答對了
```

```
0 到 1000中選一個數字：500
0 到 500中選一個數字：250
250 到 500中選一個數字：375
375 到 500中選一個數字：412
412 到 500中選一個數字：460
412 到 460中選一個數字：433
412 到 433中選一個數字：420
420 到 433中選一個數字：426
426 到 433中選一個數字：430
430 到 433中選一個數字：431
答對了
```

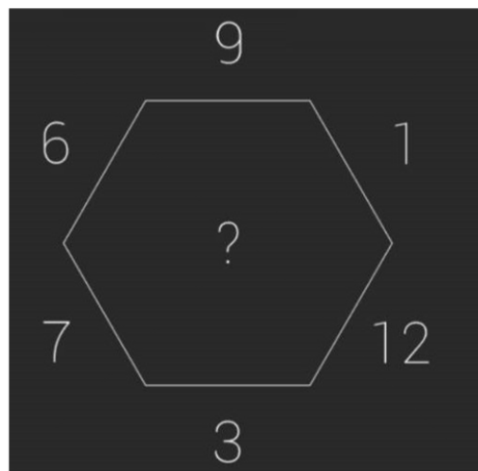
GAME

$$\begin{array}{ccccccc}
 & & & 18 & & & \\
 & & & + & & & \times \\
 & & & & - & & = \\
 & & & = & & + & = \\
 & + & 19 & = & & & - \\
 - & & + & & & + & = 33 \\
 4 & + & & = & & & = \\
 = & & = & & & & \\
 & \times & & = 25 & & &
 \end{array}$$

6 10 24 10 6 23 5 48
25 81 2 1 4 5 27

$$\begin{array}{ccccccc}
 & & - & 15 & = & & \\
 & - & & \times & & + & \\
 26 & & 21 & + & & = & \\
 = & & \div & & = & & - \\
 & - & & = & & 20 & \\
 + & & & = & & & = \\
 & & 90 & \div & = & & 13 \\
 = & & & & & + & \\
 20 & - & 3 & = & & 10 & \div & = 5 \\
 & & & & & = & & \\
 & & & & & 24 & \div & = 2
 \end{array}$$

7 7 17 11 12 9 45
2 28 2 15 13 28 2



$$\begin{array}{l}
 7, 2 = 59 \\
 12, 5 = 717 \\
 18, 9 = 927 \\
 26, 3 = ?
 \end{array}$$

3. 遊戲改良

現在這樣的程式已經是一款遊戲了，但好像有些單調。終極密碼這款遊戲可以用 2 的次方數推估需要猜的次數。例如因為 $2^{10}=1024$ ，所以今天從 0~1000 猜數字，理論上最多只需要 10 次

因次，接下來改良一下程式碼，讓每局都只有 10 次機會而已：

第 7 行 → `times=10` 代表猜 10 次

第 9 行 → 輸出 剩下 `times` 次

第 12 行 → 每次輸入一個數字，`times` 則會減一

第 23 行 → `if(times==0)` 判斷當次數等於 0 的時候就會結束遊戲

```
4 srand(time(0));
5 int number=rand()%1000;
6 int low=0,high=1000;
7 int times=10;// ←
8 for(;;){
9     cout<<"剩"<<times<<"次，從";//
10    cout<<low<<"到"<<high<<"中選一
11    int x;cin>>x;
12    times--;// ←
13    if(x==number){
14        cout<<"答對了"<<endl;
15        break;
16    }
17    if(x>number){
18        high=x;
19    }
20    if(x<number){
21        low=x;
22    }
23    if(times==0){// ←
24        cout<<"沒次數了"<<endl;//
25        break;// ←
26    }//
27 }
```

```
剩10次，從0到1000中選一個數字：500
剩9次，從0到500中選一個數字：250
剩8次，從0到250中選一個數字：125
剩7次，從125到250中選一個數字：175
剩6次，從175到250中選一個數字：220
剩5次，從220到250中選一個數字：235
剩4次，從220到235中選一個數字：228
答對了
```

```
剩10次，從0到1000中選一個數字：100
剩9次，從100到1000中選一個數字：200
剩8次，從200到1000中選一個數字：300
剩7次，從300到1000中選一個數字：400
剩6次，從400到1000中選一個數字：500
剩5次，從500到1000中選一個數字：600
剩4次，從600到1000中選一個數字：700
剩3次，從700到1000中選一個數字：800
剩2次，從800到1000中選一個數字：900
剩1次，從800到900中選一個數字：850
沒次數了
```

有比較需要動腦的玩法後，也要給玩家有重複挑戰或是直接放棄的選項

繼續 [Y] / 放棄 [N]

以下會示範那種「選項式的問答」怎麼實作：

Coding C++ RUN MENU
Auto saved at 18:09:36

```
1 #include<bits/stdc++.h>
2 using namespace std;
3 int main() {
4     srand(time(0));
5     int number=rand()%1000;
6     int low=0,high=1000;
7     char choose;
8     for(;;) {
9         cout<<"遊戲開始[Z]/退出[X]:";
10        cin>>choose;
11        if(choose=='Z') {
12            for(;;) {
13                // 這塊是終極密碼
14            }
15        }
16    }
17    if(choose=='X'){
18        cout<<"遊戲結束"<<endl;
19        break;
20    }
21 }
```

這塊是終極密碼

第7行→ `char choose` 代表，輸入後 `choose` 會記住輸入的東西

第9行→ 輸出選項讓玩家看到

第10行→輸入選項存入 `choose`

第11行→如果 `choose` 等於 `Z` 會執行12~23，執行遊戲

第25行→如果 `choose` 等於 `X` 會執行25~28，結束遊戲

GAME

	+		=	43	31	+		=	
		+		+					+
17			-		=	1			23
x		=		=		+			=
1		43				x		=	
=						-		+	
	-	11	=			+	8	=	
		-		x	=		=		
		8			+	6	=		
		=		=					
	+		=						

33	27	60	7	2	57	10	17
16	3	10	16	8	15	56	6
2	58	16					

	+	14	=						
-									
		15		+		=	32		
=		+		-		-			
11	+		=			-		=	4
	-		=		=		+		
		-		=	2		1		
=							=		
19	-	5	=				4		

12	3	7	3	14	9
9	23	5	20	23	37

$$A - 4B + 2C = 16$$

$$2A + 7B - 5C = 83$$

$$A + B - C = ?$$