

## E94116075 駱穎玫 Hw2 程式作業

### 1. 實現方法

#### (1)Hw2-1:

利用亂數產生器重複選出介於 0~1 的小數，設一個變數 sum 為抽到的總次數，若此小數小於 0.006(則代表抽卡抽到)，則將此次紀錄+1，最後將抽到的總次數除選出小數的次數等於模擬的機率。

#### (2)Hw2-2:

除了上述的作法，多加一個紀錄次數的變數 count，紀錄當下的抽數是循環裡的第幾抽，並依據 count 等於多少來設定機率。

### 2. 實驗分析與結果討論

#### (1)Hw2-1:

A. 在抽取次數越大對抽籤機率精準度的影響

```
int main(void)
{
    int sum=0; //record the times the user wins the lottery
    srand((unsigned)time(NULL)); // initialize random number generator
    for(int i=0; i<10; i++){ //run 1000000 times to raise the precision of the data
        float ans;

        ans=(float)rand()/RAND_MAX; // produce a float number between 0~1 in random
        if(ans<=0.006){
            sum=sum+1;
        }
    }
    float answer;
    answer=0.1*sum*100;
    printf("抽中五星角色的機率為 %.2f%%\n", answer);

    int sum1=0; //record the times the user wins the lottery
    for(int i=0; i<10; i++){ //run 1000000 times to raise the precision of the data
        float ans1;

        ans1=(float)rand()/RAND_MAX; // produce a float number between 0~1 in random
        if(ans1<=0.051){
            sum1=sum1+1;
        }
    }
    float answer1;
    answer1=0.1*sum1*100;
    printf("抽中四星角色的機率為 %.2f%%", answer1);
    return 0;
}
```

把圈取來的地方分別改成

- $i < 10$

```
抽中五星角色的機率為 10.00%
抽中四星角色的機率為 10.00%
-----
```

- $i < 1000$

抽中五星角色的機率為 1.10%  
抽中四星角色的機率為 5.60%

- $i < 10000$

抽中五星角色的機率為 0.44%  
抽中四星角色的機率為 5.69%

- $i < 100000$

抽中五星角色的機率為 0.59%  
抽中四星角色的機率為 5.00%

- $i < 1000000$  (10 的 6 次方)

抽中五星角色的機率為 0.61%  
抽中四星角色的機率為 5.13%

- $i < 10000000$  (10 的 7 次方)

抽中五星角色的機率為 0.60%  
抽中四星角色的機率為 5.09%

- $i < 100000000$  (10 的八次方)

抽中五星角色的機率為 0.60%  
抽中四星角色的機率為 5.10%

- 結果討論：

- 抽取次數越多，所模擬出的機率會趨於官方給的抽卡機率。
- 抽取次數越多，所模擬出的機率越趨於一致。
- 大約在迴圈跑 10 的八次方以上時，機率會幾乎等同於官方給的抽卡機率。

(2)Hw2-2:

A. 我在第二題模擬出 12.1%的機率，發現同學都成功模擬出 13.0X%的機率，於是開始尋找自己到底是哪裡寫錯。


```
int main(void)
{
    int sum1=0,count1=0; //record the times the user wins the Lottery and the times relative to the odd
    srand((unsigned)time(NULL)); // initialize random number generator
    for(int j=0;j<1000000;j++){ //run 1000000 times to raise the precision of the data
        float ans1;
        ans1=(float)rand()/RAND_MAX; // produce a float number between 0~1 in radom
        if(count1<=8&&ans1<=0.051){//if the count1 is <= 8 and ans1 is <=0.051
            sum1+=1;
            count1=0;
        }
        else if(count1==9&&ans1<=0.562){
            sum1+=1;
            count1=0;
        }
        else if(count1==10&&ans1<=0.994){
            sum1+=1;
            count1=0;
        }
        else if(count1>10){
            sum1+=1;
            count1=0;
        }
        else{
            count1+=1;
        }
    }
    float answer1;
    answer1=0.0000001*sum1*100;
    printf("抽中四星角色(含保底)的機率為 %.2f%%",answer1);
    return 0;
}
```

- 我忽略 count+=1 這行必須重複執行(不管有沒有抽到)，如果依照我原本錯誤的方法，在抽到的時後就不會記錄 count 的次數，所以會造成誤差。

## B. 用亂數抽取數字的範圍設定

```
#include <stdio.h>
#include <stdlib.h>
#include <time.h> // remember to "include" the time function when using it

int main(void)
{
    int sum1=0, count1=0; //record the times the user wins the Lottery and the times relative to the odd
    srand((unsigned)time(NULL)); // initialize random number generator
    for(int j=0; j<10; j++){ //run 1000000 times to raise the precision of the data
        int ans1;
        int max=100, min=1;
        ans1=(int)rand()%(max-min+1)+1; // produce a float number between 0~1 in radom
        printf("%d\n", ans1);
    }
    return 0;
}
```



```
1
19
97
74
31
21
89
80
79
37

-----
Process exited after 0.02772 seconds with return value 0
請按任意鍵繼續 . . .
```

- 將亂數生產器初始化後產生一個數 ans1 令他等於 rand()%(最大值-最小值)+1，其中 rand 為一個隨機數，取其餘數+1 可隨機取得一個最大值及最小值間的數
- 需注意的是 rand 的變數型態應與 ans1 相同